



เครื่องเติมลมอัตโนมัติควบคุมโดยไมโครคอนโทรลเลอร์

AUTOMATIC AIR COMPRESSOR WITH MICROCONTROLLER

นายชเนต สามลี รหัส 53362778

นายศุภวิชญ์ ตั้งชัยก รหัส 53363102

ห้องปฏิบัติการวิศวกรรมศาสตร์
รับที่รับ..... 19 / ๗.๑. 57
เลขทะเบียน..... 16899067
เลขเรียกหนังสือ..... ๗5.
มหาวิทยาลัยเกษตรศาสตร์ ๒๕๕๖

i 689906

2556

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

ปีการศึกษา 2556



ใบรับรองปริญญาโท

ชื่อหัวข้อโครงการ เครื่องเติมลมอัด โนมติกควบคุม โดยไมโครคอนโทรลเลอร์
ผู้ดำเนินโครงการ นายธนศ สามสี รหัส 53362778
นายศุภวิชญ์ สังข์ยก รหัส 53363102
ที่ปรึกษาโครงการ ดร. มุขิตา สงฆ์จันทร์
สาขาวิชา วิศวกรรมไฟฟ้า
ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา 2556

.....
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยธนเรศวร อนุมัติให้ปริญญาโทฉบับนี้เป็นส่วนหนึ่ง
ของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

.....
.....ที่ปรึกษาโครงการ
(ดร. มุขิตา สงฆ์จันทร์)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. ศุภวรรณ พลพิทักษ์ชัย)

.....กรรมการ
(อาจารย์เศรษฐา ตั้งถ้วานิช)

ชื่อหัวข้อโครงการ เครื่องเติมลมอัด โนมติควบคุมโดยไมโครคอนโทรลเลอร์
ผู้ดำเนินโครงการ นายธนศ สามสี รหัส 53362778
นายศุภวิชญ์ สังข์ชัยก รหัส 53363102
ที่ปรึกษาโครงการ ดร. มุทิตา สงฆ์จันทร์
สาขาวิชา วิศวกรรมไฟฟ้า
ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา 2556

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอเครื่องเติมลมอัด โนมติโดยใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุมการทำงานเพื่อใช้เติมลมได้สะดวกและสามารถพกพาได้ โดยการเติมนั้นสามารถสั่งการผ่านการควบคุมด้วยมือและผ่านการควบคุมด้วยคอมพิวเตอร์ ซึ่งจากการทดลองการทำงานของเครื่องเติมลมอัด โนมติพบว่าเครื่องเติมลมสามารถใช้กับรถยนต์ได้ทุกรุ่นทุกยี่ห้อ โดยสามารถเติมลมได้แรงดันสูงสุด 40 ปอนด์ต่อตารางนิ้ว (pond per square inch: PSI)

Project title Automatic Air Compressor with Microcontroller
Name Mr. Thanet Samsee ID. 53362778
Mr. Supawich Sangyok ID. 53363102
Project advisor Ms. Mutita Songjun, Ph.D.
Major Electrical Engineering
Department Electrical and Computer Engineering
Academic year 2013

Abstract

This project presents the automatic vacuum pump controlled by microcontroller. The advantage of the proposed automatic vacuum pump it is able to fill a tyre conveniently and it is very easy to carry. The operation of the automatic vacuum pump is controlled by manual and also by computer via visual basic program. The experiments show that this automatic vacuum pump can be used with all type of family car. And it produces the pressure up to 40 PSI.

กิตติกรรมประกาศ

โครงการนี้สำเร็จลุล่วงไปได้ด้วยความกรุณาเป็นอย่างยิ่งจาก ดร. มุทิตา สงฆ์จันทร์ ซึ่งเป็นที่ปรึกษาโครงการและให้ความกรุณาในการเอาใจใส่ในรายละเอียดในการสร้างชิ้นงานและตรวจทานปริญญาานิพนธ์รวมถึงการให้คำแนะนำและให้คำปรึกษาในการแก้ไขปัญหาตลอดการดำเนินงาน

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร. ศุภวรรณ พลพิทักษ์ชัยและอาจารย์เศรษฐา ตั้งค้ำวานิช ที่ให้เกียรติเป็นกรรมการในการตรวจสอบโครงการ ที่ให้คำปรึกษาและชี้แนะแนวทางในการดำเนินโครงการทำให้โครงการมีความสมบูรณ์แบบยิ่งขึ้น

ขอขอบพระคุณคณาจารย์ทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้ให้กับผู้ดำเนินโครงการ การให้คำแนะนำและการช่วยเหลือจนทำให้โครงการนี้สำเร็จลุล่วงไปได้ด้วยดี

เหนือสิ่งอื่นใดคณะผู้ดำเนินโครงการขอกราบขอบพระคุณบิดามารดา ผู้มอบความรักความเมตตา สติปัญญา รวมทั้งเป็นผู้ให้ทุกสิ่งทุกอย่างตั้งแต่วัยเยาว์จวบจนถึงปัจจุบัน คอยเป็นกำลังใจทำให้ได้รับความสำเร็จอย่างทุกวันนี้ และขอขอบคุณทุกๆคนในครอบครัวของคณะผู้ดำเนินโครงการที่ไม่ได้กล่าวไว้ ณ ที่นี้ ซึ่งให้กำลังใจและการสนับสนุนในทุกๆด้านอย่างดีที่สุดเสมอมา

นายธนศ

สามสี

นายศุภวิชญ์

สังข์ยก

สารบัญ

	หน้า
ใบรับรองปริญญาโท..... ก	ก
บทคัดย่อภาษาไทย..... ข	ข
บทคัดย่อภาษาอังกฤษ..... ค	ค
กิตติกรรมประกาศ..... ง	ง
สารบัญ..... จ	จ
สารบัญตาราง..... ช	ช
สารบัญรูป..... ฉ	ฉ
บทที่ 1 บทนำ..... 1	1
1.1 ที่มาและความสำคัญของโครงการ..... 1	1
1.2 วัตถุประสงค์ของโครงการ..... 1	1
1.3 ขอบเขตของโครงการ..... 1	1
1.4 แนวทางการดำเนินงาน..... 1	1
1.5 แผนการดำเนินงาน..... 2	2
1.6 ประโยชน์ที่คาดว่าจะได้รับ..... 2	2
1.7 งบประมาณที่ใช้..... 3	3
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง..... 4	4
2.1 หลักการลม..... 4	4
2.1.1 ฟิสิกส์ของลม..... 5	5
2.2 การตรวจวัดความดัน..... 6	6
2.3 นิวแมติกส์..... 6	6
2.3.1 การใช้อากาศอัดเป็นตัวกลาง..... 6	6
2.3.2 ส่วนประกอบของระบบนิวแมติกส์..... 8	8
2.3.3 ข้อดีของลมอัด..... 9	9
2.3.4 ข้อเสียของลมอัด..... 9	9
2.4 เซนเซอร์ (sensor)..... 10	10
2.4.1 เซนเซอร์ตรวจวัดความดัน..... 11	11

สารบัญ (ต่อ)

	หน้า
2.5 ไมโครคอนโทรลเลอร์	13
2.5.1 ความเร็วของไมโครคอนโทรลเลอร์ PIC 16F877	13
2.5.2 หน่วยความจำของไมโครคอนโทรลเลอร์ PIC 16F877.....	14
2.5.3 คุณสมบัติของไมโครคอนโทรลเลอร์ PIC 16F877.....	14
2.5.4 วงจรภายในไมโครคอนโทรลเลอร์.....	15
2.6 โปรแกรมไมโครซอฟท์วิซวลเบสิก (Microsoft Visual Basic)	15
2.6.1 เริ่มต้นใช้งานโปรแกรม (Microsoft Visual Basic).....	16
2.7 โซลินอยด์วาล์ว.....	17
2.7.1 วาล์วปกติปิดเลื่อนวาล์วกลับด้วยแรงสปริง.....	17
2.7.2 วาล์วเลื่อนกลับด้วยโซลินอยด์วาล์ว	18
บทที่ 3 วิธีการดำเนินโครงการ.....	19
3.1 ระบบสั่งการ	19
3.1.1 ระบบสั่งการควบคุมด้วยมือ (MANUAL)	20
3.1.2 ระบบสั่งการด้วยคอมพิวเตอร์ (COMPUTER).....	21
3.2 ระบบประมวลผลคำสั่ง	25
3.2.1 วงจรไมโครคอนโทรลเลอร์	25
3.2.2 วงจรสวิตช์ควบคุมแรงดัน	26
3.2.3 วงจรเซนเซอร์ตรวจวัดแรงดัน	27
3.2.4 วงจรควบคุมโซลินอยด์วาล์ว.....	28
3.2.5 วงจรควบคุมเครื่องปั๊มลม	28
3.2.6 วงจรภายในหน้าจอแสดงผลแบบดิจิทัล	29
3.2.7 วงจรไอซีเบอร์ MAX232.....	29
3.2.8 วงจรแบตเตอรี่	30
3.3 โครงสร้างเครื่องเติมลมอัตโนมัติ	31
บทที่ 4 ผลการทดลองและผลการวิเคราะห์.....	32
4.1 การทดลองความสามารถในการเติมลมผ่านการควบคุมด้วยมือ.....	32

สารบัญ (ต่อ)

	หน้า
4.2 การทดสอบความสามารถในการเติมลมผ่านการควบคุมด้วย โปรแกรมวิซวลเบสิก	34
4.2.1 แบบอัตโนมัติ	34
4.2.2 แบบป้อนค่า.....	36
4.3 การทดสอบระยะเวลาที่ใช้ในการเติมลม.....	38
บทที่ 5 สรุปผลและข้อเสนอแนะ.....	40
5.1 สรุปผลการทดลองการทำงานของเครื่องเติมลมอัตโนมัติ.....	40
5.2 ปัญหาและการแก้ไขปัญหา	40
5.3 แนวทางในการพัฒนาโครงการ.....	41
เอกสารอ้างอิง	42
ภาคผนวก ก โปรแกรมการทำงานของเครื่องเติมลมอัตโนมัติ.....	43
ภาคผนวก ข โปรแกรมการทำงานของโปรแกรมวิซวลเบสิก	52
ภาคผนวก ค รายละเอียดของไมโครคอนโทรลเลอร์ PIC 16F877.....	66
ภาคผนวก ง รายละเอียดเกี่ยวกับ MPX 5700.....	79
ประวัติผู้ดำเนินโครงการ.....	84

สารบัญตาราง

ตารางที่	หน้า
1.1 แผนการดำเนินงาน	2
4.1 แสดงผลการทดลองความสามารถในการเติมลมผ่านการควบคุมด้วยมือ	33
4.2 แสดงผลการทดลองความสามารถในการเติมลมผ่านการควบคุมด้วย โปรแกรมวิซวลเบสิกแบบอัตโนมัติ.....	35
4.3 แสดงผลการทดลองความสามารถในการเติมลมผ่านการควบคุมด้วย โปรแกรมวิซวลเบสิกแบบป้อนค่า	34
4.4 แสดงผลการทดลองระยะเวลาที่ใช้ในการเติมลม	38



สารบัญรูป

รูปที่	หน้า
2.1 ลักษณะของความเร็วลมภายใต้ชั้นบรรยากาศ (atmosphere boundary layer)	5
2.2 เซนเซอร์ตรวจวัดความดัน	11
2.3 วัดความดันของไหลจากอุณหภูมิจาก.....	12
2.4 กราฟแสดงให้เห็นถึงความสัมพันธ์ของขนาดความยาวของท่อกับอุณหภูมิ.....	13
2.5 รูปแสดงการทำงานวาล์ว แบบเลื่อนวาล์วกลับด้วยแรงสปริง.....	17
2.6 รูปแสดงการทำงานวาล์ว แบบวาล์วเลื่อนกลับด้วยโซลินอยด์วาล์ว	18
3.1 แผนภาพแสดงภาพรวมของระบบ (System Overview Diagram).....	19
3.2 การทำงานของเครื่องเติมลมอัตโนมัติ.....	20
3.3 แสดงปุ่มรับคำสั่งการควบคุมด้วยมือ.....	21
3.4 รูปแสดงปุ่มรับคำสั่งการควบคุมด้วยคอมพิวเตอร์.....	22
3.5 รูปแสดงการเลือกยี่ห้อ	23
3.6 รูปแสดงการเลือกรุ่น.....	23
3.7 รูปแสดงการเลือกล้อ.....	24
3.8 รูปแสดงระบบสั่งการผ่านคอมพิวเตอร์แบบอัตโนมัติ.....	24
3.9 รูปแสดงระบบสั่งการผ่านคอมพิวเตอร์แบบป้อนค่า.....	25
3.10 วงจรไมโครคอนโทรลเลอร์.....	26
3.11 วงจรสวิตช์ควบคุมแรงดัน	26
3.12 วงจรเซนเซอร์ตรวจวัดแรงดัน	27
3.13 วงจรควบคุมโซลินอยด์วาล์ว	28
3.14 วงจรควบคุมเครื่องปั๊มลม.....	28
3.15 วงจรหน้าจอแสดงผลแบบดิจิตอล	29
3.16 วงจรไอซีเบอร์ MAX232	30
3.17 วงจรแบตเตอรี่	30
3.18 รูปแสดงโครงสร้างเครื่องเติมลม	31
4.1 รูปแสดงการเติมลมผ่านการควบคุมด้วยมือ.....	32
4.2 รูปแสดงการเติมลมผ่านการควบคุมด้วยโปรแกรมวิซวลเบสิก.....	34
4.3 รูปแสดงลักษณะการทำงานของโปรแกรมแบบอัตโนมัติ.....	35
4.4 รูปแสดงลักษณะการทำงานของโปรแกรมแบบป้อนค่า	36

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ปัจจุบันในงานอุตสาหกรรมบางประเภทที่มีการนำระบบควบคุมแรงดันของลมหรือก๊าซมาใช้ในขบวนการผลิตของโรงงานอุตสาหกรรมมีอย่างแพร่หลายเช่นอุตสาหกรรมพลาสติก อุตสาหกรรมการผลิตขวดแก้วอุตสาหกรรมบรรจุก๊าซ เป็นต้น ซึ่งได้นำระบบควบคุมแรงดันมาใช้เพื่อควบคุมให้ผลิตภัณฑ์อยู่ในมาตรฐานที่กำหนด นอกจากนี้ระบบควบคุมแรงดันยังสามารถที่จะนำมาประยุกต์ใช้กับยานพาหนะทุกประเภทได้จึงเป็นแนวทางในการสร้างเครื่องเติมลมอัตโนมัติเพื่อนำไปประยุกต์ใช้ในการเติมลมยางของยานพาหนะเพื่อให้เกิดความสะดวกสบาย และได้มาตรฐานในการเติมลม ซึ่งการเติมลมยางให้ได้ตามมาตรฐานของยานพาหนะแต่ละประเภทจะช่วยเพิ่มความปลอดภัยในการขับขี่ยานพาหนะบนท้องถนนและยังช่วยลดการใช้เชื้อเพลิงซึ่งสอดคล้องกับนโยบายของรัฐบาลที่ส่งเสริมและรณรงค์ลดการใช้พลังงาน

1.2 วัตถุประสงค์

สร้างเครื่องเติมลมอัตโนมัติเพื่อใช้ในการเติมลมล้อรถยนต์ควบคุมโดยใช้ไมโครคอนโทรลเลอร์ได้

1.3 ขอบเขตของโครงการ

1. เครื่องเติมลมอัตโนมัติสามารถเติมลมได้แรงดันสูงสุดเท่ากับ 40 ปอนด์ต่อตารางนิ้ว ซึ่งเท่ากับแรงดันที่ใช้เติมลมล้อรถยนต์
2. เครื่องเติมลมอัตโนมัติสามารถแสดงผลผ่านหน้าจอคอมพิวเตอร์ได้โดยโปรแกรมวิซวลเบสิก (Visual basic)

1.4 แนวทางการดำเนินงาน

1. ศึกษาการทำงานของเซนเซอร์ตรวจวัดแรงดันและโซลินอยด์วาล์ว
2. ศึกษาการทำงานของโปรแกรมภาษาซีที่ใช้กับไมโครคอนโทรลเลอร์

3. สร้างวงจรจับที่ใช้ในการควบคุมมอเตอร์และโซลินอยด์วาล์ว

4. ใช้คอมพิวเตอร์ป้อนค่าผ่าน โปรแกรมวิชวลเบสิก (Visual basic) และแสดงผลผ่านหน้าจอแสดงผล

5. ทดลองการทำงานเต็มลมในสภาพจริงและแก้ไขข้อบกพร่อง

1.5 แผนการดำเนินงาน

ตารางที่ 1.1 แผนการดำเนินงาน

กิจกรรม	เดือน/ปี 2556									
	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	
1. ศึกษาการทำงานของเซนเซอร์และโซลินอยด์วาล์ว	←		→							
2. ศึกษาการทำงานของโปรแกรมภาษาซีและไมโครคอนโทรลเลอร์	←		→							
3. ศึกษาการทำงานของวงจรจับมอเตอร์		←		→						
4. ทำการเขียนโปรแกรมควบคุมและวิชวลเบสิก(Visual basic)				←		→				
5. สร้างชุดควบคุมทั้งหมด	←		→							
6. ทดสอบการทำงานและแก้ไข						←		→		
7. สรุปผลการทดลองและจัดทำรูปเล่มปริญาานิพนธ์								←		→

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้เครื่องเต็มลมอัตโนมัติที่สามารถเต็มลมล้อยยนต์ได้
2. สามารถใช้งาน โปรแกรมวิชวลเบสิก (Visual basic)
3. สามารถนำความรู้จากเครื่องเต็มลมอัตโนมัติไปประยุกต์ใช้ในงานอุตสาหกรรมพลาสติก อุตสาหกรรมการผลิตขวดแก้วอุตสาหกรรมบรรจุภัณฑ์

1.7 งบประมาณ

1. ค่าอุปกรณ์ในการทำเครื่องเคมิลมอัตโนมัติ	1,400	บาท
2. ค่าทำรูปเล่มปริญญาบัตร	600	บาท
รวมเป็นเงินทั้งสิ้น (สองพันบาทถ้วน)	<u>2,000</u>	บาท
<u>หมายเหตุ</u> : ถัวเฉลี่ยทุกรายการ		



บทที่ 2

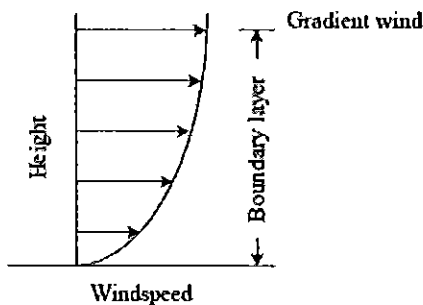
ทฤษฎีและหลักการที่เกี่ยวข้อง

โครงสร้างของเครื่องเติมลมอัดโนมัติแบ่งออกเป็น 3 ส่วนคือ ส่วนของการตรวจจับความดัน ส่วนประมวลผล และส่วนแสดงแรงดันลม โดยการสร้างเครื่องเติมลมอัดโนมัติจะคำนึงถึงความสามารถในการตรวจจับ ประมวลผล และแสดงผลที่สามารถใช้งานได้จริง

2.1 หลักการของลม

ลม [1] เกิดจากการเคลื่อนที่ของมวลอากาศซึ่งมีอุณหภูมิแตกต่างกันปรากฏการณ์นี้เกิดขึ้นโดยธรรมชาติทั่วโลก ลมที่มักจะได้ยินชื่ออยู่บ่อยๆ ก็คือ ลมมรสุม ซึ่งหมายถึงลมที่พัดเปลี่ยนทิศทางกับการเปลี่ยนฤดู คือฤดูร้อนจะพัดอยู่ในทิศทางหนึ่งและจะพัดเปลี่ยนทิศทางในทางตรงกันข้ามในฤดูหนาว นอกจากนี้ลมยังอาจเกิดขึ้นจากอิทธิพลของภูมิประเทศและความเปลี่ยนแปลงของความกดอากาศในพื้นที่นั้นๆ ซึ่งเรียกลมชนิดนี้ว่าลมประจำถิ่นซึ่งลมประจำถิ่นยังสามารถแบ่งออกเป็น ลมบกและลมทะเล ลมภูเขาและลมหุบเขา นอกจากนี้ในประเทศไทยยังมีลมประจำถิ่นที่เป็นที่รู้จักกันดีคือ ลมตะเภา และลมว่าว

กังหันลมจะใช้ประโยชน์จากลมที่อยู่ใกล้ผิวโลกหรือที่เรียกว่าลมผิวพื้นหมายถึงลมที่พัดในบริเวณผิวพื้นโลกภายใต้ความสูงประมาณ 1 กิโลเมตรเหนือพื้นดินเป็นบริเวณที่มีการคลุกเคล้าของอากาศและมีแรงฝัดอันเกิดจากการปะทะกับสิ่งกีดขวางร่วมกระทำด้วยในระดับต่ำ แต่ที่ระดับความสูงมากกว่า 10 เมตรขึ้นไปแรงต้านจะลดลงและความเร็วลมจะเพิ่มขึ้น ดังรูปที่ 2.1 ส่วนที่ระดับความสูงใกล้ 1 กิโลเมตร เกือบไม่มีแรงฝัด ความเร็วลมเปลี่ยนแปลงขึ้นอยู่กับระดับความสูงและสภาพภูมิประเทศ เช่นเดียวกันกับทิศทางของลมจากประสบการณ์ที่ผ่านมาพบว่า กังหันลมจะทำงานได้ดีหรือไม่ขึ้น ก็ขึ้นอยู่กับตัวแปรทั้งสองนี้ ที่ความเร็วลมเท่าๆกัน แต่มีทิศทางลมที่แตกต่างกัน เมื่อพุ่งเข้าหาแกนหมุนของกังหันลมแล้ว จะส่งผลต่อแรงบิดของกังหันลมเป็นอย่างมาก ผลคือแรงลัพธ์ที่ได้ออกมาจากกังหันลมแตกต่างกัน



รูปที่ 2.1 ลักษณะของความเร็วลมภายใต้ชั้นบรรยากาศ (atmosphere boundary layer) [1]

2.1.1 ฟิสิกส์ของลม

กำลังของลมที่เคลื่อนที่ด้วยความเร็ว V ผ่านพื้นที่หน้าตัด A คือ

$$P_w = \frac{1}{2} \rho A V^3 \tag{2.1}$$

เมื่อ	P_w	คือ	กำลังของลม (W)
	ρ	คือ	ความหนาแน่นของอากาศ มีค่าเท่ากับ 1.225 kg/m ³
	A	คือ	พื้นที่หน้าตัด (m ²)
	V	คือ	ความเร็วลม (m/s)

ความหนาแน่นของอากาศเป็นฟังก์ชันของความกดอากาศและอุณหภูมิที่ระดับสูงกว่าระดับน้ำทะเล ซึ่งสามารถเขียนเป็นสมการได้ดังนี้

$$\rho(z) = \frac{P_0}{(RT) \exp\left(\frac{-gz}{RT}\right)} \tag{2.2}$$

เมื่อ	$\rho(z)$	คือ	ความหนาแน่นของอากาศซึ่งเป็นฟังก์ชันของระดับความสูง (kg/m ³)
	P_0	คือ	ความดันบรรยากาศที่ระดับน้ำทะเลมาตรฐาน (kg/m ²)
	R	คือ	ค่าคงที่ของอากาศ (Specific gas constant of air) (J/K mol)
	T	คือ	อุณหภูมิ (K)
	g	คือ	ค่าแรงดึงดูดของโลก (m/s ²)
	z	คือ	ค่าความสูงจากระดับน้ำทะเล (m)

2.2 การตรวจวัดความดัน (Pressure Sensor)

อัตราการไหล (Volume flowrate, Q) [2] คือปริมาตรของของไหลซึ่งไหลผ่านท่อหรือช่องการไหลใดๆ ในหนึ่งหน่วยเวลา หรืออีกนัยหนึ่งก็คือของไหลที่ไหลผ่านพื้นที่หน้าตัดในแนวตั้งฉาก (A) ด้วยความเร็วค่าหนึ่ง (V) ซึ่งสามารถแสดงได้ดังสมการต่อไปนี้

$$Q = VA \quad (2.3)$$

อัตรามวลไหล (Mass flowrate) คือมวลของของไหลซึ่งไหลผ่านท่อหรือช่องการไหลใดๆ ในหนึ่งหน่วยเวลา

$$\dot{m} = \rho Q = \rho VA \quad (2.4)$$

2.3 นิวแมติกส์

ระบบนิวแมติกส์ [3] หมายถึง ระบบทำงาน โดยใช้อากาศเป็นตัวส่งกำลังในการขับเคลื่อนอุปกรณ์ทำงานของเครื่องจักรต่างๆ เช่น กระบอกลูกสูบ หรือมอเตอร์ลม เป็นต้น

นิวแมติกส์ (pneumatic) มาจากคำว่า นิวมา (pneuma) เป็นภาษากรีกโบราณหมายถึง ลม หรือลมหายใจทางปรัชญาหมายถึง วิญญาณเป็นการศึกษาเกี่ยวกับลมและลมที่เคลื่อนที่ลมอัดจึงเป็นพลังงานเก่าแก่ที่มนุษย์รู้จักนำมาใช้ประโยชน์เป็นเวลาหลายพันปีมาแล้วประมาณ 3,000 ปีก่อนคริสต์ศักราช มนุษย์ได้รู้จักวิธีการถลุงแร่ ทองคำ ทองแดง ดีบุก และใช้ลูกสูบเป่าไฟในการช่วยถลุงแร่ ต่อมาทฤษฎีของ (Ktesibios) ชาวกรีกโบราณได้สร้างปืนใหญ่โดยใช้ลมอัดเป็นตัวส่งกำลังเมื่อ 2,000 กว่าปีมาแล้วและคนป่าได้ใช้ลมอัดเป่าลูกดอกจากกระบอกลมไม้ไผ่สำหรับหาอาหารหรือป้องกันตัว

2.3.1 การใช้อากาศอัดเป็นตัวกลาง

เนื่องจากระบบอัตโนมัติ โดยเฉพาะในระบบอัตโนมัติเล็กๆ มีปัญหาบางอย่างทำให้พบว่าไม่มีตัวกลางการทำงานใดที่สามารถนำมาใช้ได้ง่ายกว่าและประหยัดกว่าระบบนิวแมติกส์ดังนั้นข้อดี ของระบบนี้อาจแบ่งหัวข้อได้ดังต่อไปนี้

1. อากาศมีปริมาณที่ไม่จำกัดในทุกๆ แห่ง
 2. อากาศอัดสามารถส่งผ่านไปตามท่อได้ง่าย แม้ว่าจะมีระยะทางไกลก็ตาม
 3. อากาศอัดสามารถกักเก็บไว้ได้
 4. อากาศอัดไม่มีความจำเป็นจะต้องส่งกลับมาสามารถระบายทิ้งไปในบรรยากาศได้หลังจากการใช้งานแล้ว
 5. อากาศอัดไม่มีความรู้สึกไวต่อความเบี่ยงเบนของอุณหภูมิซึ่งทำให้การทำงานมีความแน่นอนสูงแม้ว่าจะอยู่ในสถานะอุณหภูมิสูงมากๆ ก็ตาม
 6. อากาศอัดไม่เกิดการระเบิดง่ายดังนั้นจึงไม่มีความจำเป็นต้องมีอุปกรณ์พิเศษราคาแพงอื่นๆ เพื่อใช้ป้องกันการระเบิด
 7. อากาศมีความสะอาดในระดับหนึ่งซึ่งมีความสำคัญมาก โดยเฉพาะในกรณีอุตสาหกรรมที่เกี่ยวข้องกับอาหาร การทอผ้า ผ้าขนสัตว์ และเครื่องหนัง
 8. ชิ้นส่วนของการทำงานสำหรับอากาศอัดเป็นแบบง่ายๆ และมีราคาถูกในการสร้าง
 9. อากาศอัดมีความเร็วสูงดังนั้นอัตราความเร็วของการทำงานจะสูงด้วย
 10. ความเร็วและแรงของอุปกรณ์ทำงานในระบบนิวแมติกส์สามารถปรับแต่งได้ตามต้องการ
 11. เครื่องมือและอุปกรณ์ทำงานระบบนิวแมติกส์สามารถป้องกันงานเกินกำลังได้
 12. การเคลื่อนที่ในทางตรงสามารถทำงานได้โดยตรง
- นอกจากข้อดีดังกล่าวมาแล้วนั้นระบบที่ใช้อากาศอัดก็มีข้อเสียเช่นเดียวกัน ดังตัวอย่างดังต่อไปนี้
1. อากาศอัดเป็นตัวกลางที่ค่อนข้างแพง เมื่อเปรียบเทียบกับการแปลงของพลังงานอื่นๆ (อย่างไรก็ตามจะถูกชดเชยจากราคาของอุปกรณ์ที่มีราคาถูกและเป็นแบบง่ายๆ)
 2. อากาศอัดต้องการสถานะแวดล้อมที่ดีต้องไม่มีฝุ่นหรือความชื้น
 3. เป็นไปได้ยากที่จะได้ความเร็วของกระบอกสูบให้มีค่าคงที่สม่ำเสมอ
 4. อากาศอัดจะประหยัดเฉพาะที่ใช้แรงขยายถึงจุดหนึ่งเท่านั้น โดยปกติแล้วใช้ความดันที่ 600 กิโลปาสกาล (6 บาร์) ข้อจำกัดของแรงอยู่ที่ประมาณ 20,000-30,000 นิวตันขึ้นอยู่กับความเร็วยและระยะทางที่ทำงานถ้าต้องการแรงมากกว่านี้ควรใช้ระบบไฮดรอลิก
 5. การระบายออกของอากาศมีเสียงดัง (ปัจจุบันมีทางแก้ไขคือ ใช้อุปกรณ์เก็บเสียง (silencer))

6. ละอองของน้ำมันผสมกับอากาศที่ใช้สำหรับหล่อลื่นในระบบนิวแมติกส์จะถูกระบายออกสู่บรรยากาศ ทำให้เกิดมลภาวะ (ปัจจุบันสามารถหลีกเลี่ยงได้ โดยใช้อุปกรณ์ชนิดที่ไม่ต้องการสารหล่อลื่น (non-lubricated))

2.3.2 ส่วนประกอบของระบบนิวแมติกส์

1. ต้นกำลัง คือ มอเตอร์หรือเครื่องยนต์
2. เครื่องอัดอากาศ (compressor) ทำหน้าที่เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานแรงดัน (ลมอัด) ที่มีความดันสูง
3. เครื่องระบายความร้อนลมอัด (heat exchange) ทำหน้าที่ระบายความร้อนลมอัดก่อนนำไปใช้งานเนื่องจากอากาศ ที่ถูกอัดให้มีความดันสูงจะทำให้อุณหภูมิอากาศสูงขึ้นตามไปด้วย
4. เครื่องกรองลมท่อส่งลมอัด (main air filter) ทำหน้าที่กรองลมอัดก่อนนำไปใช้งานเนื่องจากอากาศมีความชื้นและฝุ่นละออง
5. ถังเก็บลมอัดทำหน้าที่เก็บกักลมที่ทำการอัดและจ่ายลมออกด้วยความดันสม่ำเสมอ
6. เครื่องทำอากาศแห้ง (air dryer) ทำหน้าที่กำจัดความชื้นออกจากลมอัดป้องกันการเกิดหยดน้ำกลั่นตัวในระบบซึ่งจะทำความเสียหายให้อุปกรณ์อื่นได้
7. อุปกรณ์กรองลม (air filter) ทำหน้าที่คล้ายเครื่องกรองลมท่อส่งลมอัด
8. ชุดควบคุมและปรับคุณภาพลมอัด (service unit) ติดตั้งใกล้กับอุปกรณ์ทำงานหรือเครื่องจักร เพื่อกรองความชื้น ปรับความดันของลมอัด และผสมน้ำมันหล่อลื่นก่อนใช้งานประกอบด้วย
 - อุปกรณ์กรองลม (air filter) ทำหน้าที่กรองลมให้สะอาดและดักความชื้น
 - อุปกรณ์ควบคุมความดันลมอัด (pressure regulating valve) ทำหน้าที่รักษาความดันใช้งานให้อยู่คงที่ ถึงแม้ความดันต้นทางจะเปลี่ยนแปลง
 - อุปกรณ์ผสมน้ำมันหล่อลื่น (oil lubricator) ทำหน้าที่ผสมน้ำมันหล่อลื่นให้กับลมอัดเพื่อป้องกันการเสียดสีของอุปกรณ์ทำงานที่มีการเคลื่อนที่ในระบบ
9. อุปกรณ์ควบคุมทิศทางการลมอัด ได้แก่ วาล์วชนิดต่างๆ ทำหน้าที่เปลี่ยนทิศทางการเคลื่อนที่หรือเปลี่ยนทิศทางการทำงานของระบบประกอบด้วย
 - วาล์วควบคุมทิศทางการทำหน้าที่ควบคุมลูกสูบเคลื่อนที่เข้าหรือเคลื่อนที่ออก

- วาล์วปรับความเร็วทำหน้าที่ควบคุมลมอัดให้มีปริมาณมากน้อยตามความต้องการ ทำให้ถูกสูบเคลื่อนที่ช้าหรือเร็ว ได้แก่ วาล์วปรับอัตราการไหลและวาล์วคายไอเสีย

10. อุปกรณ์ทำงาน (working element) จะทำหน้าที่เปลี่ยนพลังงานแรงดัน (ลมอัด) เป็นพลังงานกล ได้แก่ กระบอกสูบ มอเตอร์ลม

11. อุปกรณ์เก็บเสียงหรือตัวเก็บเสียง (air silencer) ทำหน้าที่กรองเสียงลมหรือเก็บเสียงลมอัดที่ออกจากกระบอกสูบทิ้งไม่ให้มีเสียงดัง

2.3.3 ข้อดีของลมอัด

1. ลมอัดสะอาดและมีความปลอดภัยหากมีการรั่วก็ไม่ใช่อันตรายสามารถปล่อยสู่บรรยากาศได้โดยไม่มีผลเสียต่อสิ่งแวดล้อมและไม่ก่อให้เกิดอันตราย

2. มีปริมาณไม่จำกัด

3. การเก็บลมอัดไว้ในถังทำให้สามารถใช้งานได้ตามต้องการและอุปกรณ์ทำงานได้อย่างต่อเนื่อง

4. ไม่เกิดการระเบิดหรือติดไฟกรณีมีการรั่วซึมทำให้ไม่ต้องมีอุปกรณ์ป้องกันราคาแพง

5. อุณหภูมิใช้งานสูง

6. อุปกรณ์มีโครงสร้างง่าย ราคาถูก ทนทาน ซ่อมบำรุงรักษาง่าย

7. สามารถส่งถ่ายได้ระยะทางไกลๆ ไม่ต้องมีท่อลมกลับสามารถปล่อยทิ้งได้เลยเมื่อใช้แล้ว

8. สามารถควบคุมความเร็ว ความดัน ด้วยอุปกรณ์ที่ง่ายและราคาถูก

9. สามารถใช้งานเกินกำลัง (over load) โดยอุปกรณ์ไม่เกิดความเสียหาย

10. ระบบสามารถทำให้อุปกรณ์ทำงาน (ลูกสูบ) สามารถเคลื่อนที่ด้วยความเร็วสูงได้ประมาณ 1-2 เมตร/วินาที หรือ 10 เมตร/วินาที สำหรับลูกสูบแบบพิเศษ

2.3.4 ข้อเสียของลมอัด

1. ลมอัดมีความชื้นและฝุ่นละอองจึงต้องมีอุปกรณ์กรองความชื้นและฝุ่นละอองก่อนนำไปใช้งาน

2. ลมอัดมีเสียงดังขณะระบายทิ้งจึงต้องมีอุปกรณ์เก็บเสียง

3. ความดันของลมอัดเปลี่ยนแปลงตามอุณหภูมิทำให้ลูกสูบเคลื่อนที่ไม่สม่ำเสมอ

4. ลมอัดทำงานได้ที่ความดัน ประมาณ 7 บาร์ หรือได้แรงในช่วง 20,000-30,000 นิวตัน

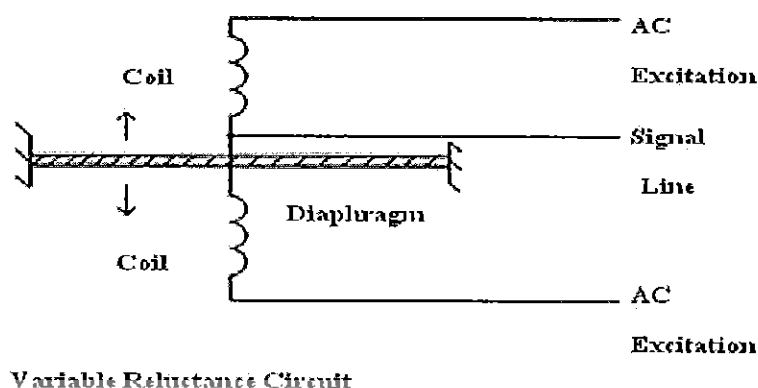
5. ลมอัดเป็นตัวกลางที่ราคาแพงเมื่อเปรียบเทียบกับระบบเปลี่ยนแปลงพลังงานอื่นๆ แต่ก็ได้รับการชดเชยจากราคาของอุปกรณ์บางชิ้น ที่มีราคาถูกและมีสมรรถนะ (จำนวนรอบของการทำงานที่สูงกว่า)

2.4 เซนเซอร์ (sensor)

เซนเซอร์ [4] เป็นอุปกรณ์ที่ทำหน้าที่ตรวจจับความรู้สึกร่างกายภาพตัวใดตัวหนึ่งแล้วเปลี่ยนค่าตัวแปรทางด้านเอาต์พุตให้เป็นตัวแปรทางด้านกายภาพตัวอื่นคุณลักษณะของเซนเซอร์ที่ต้องการหากอุปกรณ์ที่ใช้ตรวจจับสัญญาณไม่มีคุณภาพที่ดีแล้วระบบก็จะไม่สามารถทำงานเพื่อให้ผลิตภัณฑ์ออกมาดีได้จึงต้องมีขอบเขตของเซนเซอร์ที่ดีเพื่อใช้ในการปฏิบัติการดังนี้

1. มีความสามารถในการรับสัญญาณและตรวจจับพร้อมทั้งรับความรู้สึกร่างกายอินพุตที่ต้องการและต้องไม่มีความไวกับสัญญาณอื่นที่ต้องการ
2. ไม่ดัดแปลงหรือแก้ไขค่านัยสำคัญที่จะวัด
3. มีความสามารถต่อการดัดแปลงระบบ (amenable) คือให้ง่ายต่อการปรับปรุงระบบโดยใช้อุปกรณ์ที่อยู่ภายในกระบวนการเดิมๆ ได้
4. ต้องมีความแม่นยำ (accuracy) ที่ดี
5. ต้องมีความสามารถในการทำซ้ำ (reproducibility) ที่ดี
6. ต้องมีขนาด (amplitude) ที่เป็นเชิงเส้น
7. ต้องมีผลตอบสนองต่อความถี่เพียงพอ
8. ไม่สร้างหรือกำเนิดการรบกวนเฟสต่อกัน
9. ต้องมีความสามารถในการต่อต้านสิ่งรบกวนจากสิ่งแวดล้อม โดยไม่ทำลายความถูกต้องของระบบ
10. มีผลตอบสนองและความน่าเชื่อถือที่ดี

2.4.1 เซนเซอร์ตรวจวัดความดัน (Pressure Sensor)



รูปที่ 2.2 เซนเซอร์ตรวจวัดความดัน [4]

จากรูปที่ 2.2 การวัดความดันที่มีค่าต่ำมากๆ จะให้ผลค่อนข้างคลาดเคลื่อนซึ่งต้องปรับปรุงขอบเขตความสามารถในการวัดรวมไปถึงผลตอบสนองต่อความถี่สูง (High frequency response) และความทนทาน ซึ่งทั้งหมดที่ขึ้นอยู่กับขั้นตอนและกรรมวิธีในการผลิตส่วนที่ใช้วัดความดันซึ่งไม่มีส่วนเชื่อมต่อกับอุปกรณ์ส่วนอื่น โดยมีหลักดังนี้

ตัวแปรที่ใช้วัดการเปลี่ยนแปลงของความดัน อธิบายได้โดยดูจากการนำไฟฟ้าของส่วนที่เป็นไดอะแฟรม วัดความดันกับขดลวดทั้ง 2 ขดลวดถูกพันและติดตั้ง โดยมีแนวแกนทั้ง 2 กับระนาบของไดอะแฟรม ขดลวดทั้ง 2 ต่อกัน ส่วนไดอะแฟรมจะสามารถเคลื่อนที่ได้อิสระเพื่อตอบสนองผลของการเปลี่ยนแปลงความดัน

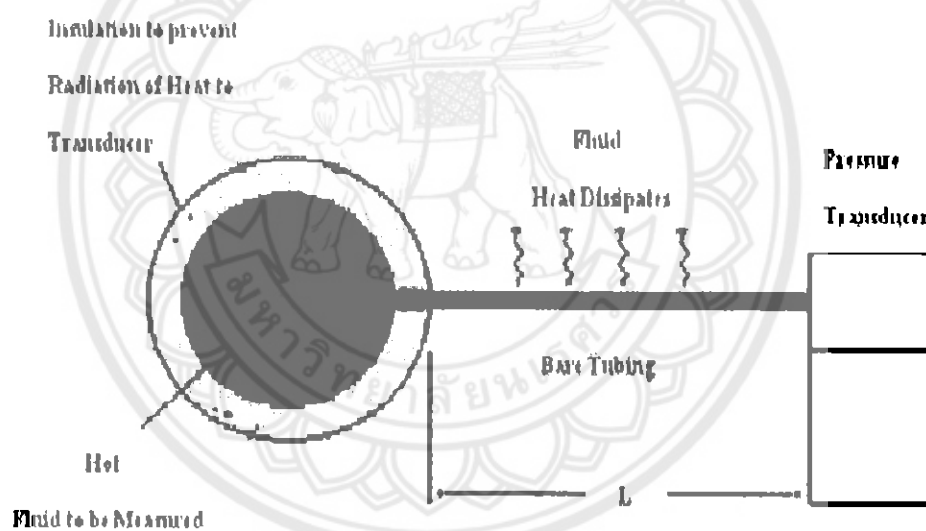
ขดลวดได้รับกระแสสลับขนาด 5 โวลต์ ที่มีความถี่ 3 หรือ 4 กิโลเฮิรตซ์ขดลวดทั้งต่อเข้าสายสัญญาณสายเดียวกัน จึงถือว่าไม่มีความต้านทานความดันไดอะแฟรม จะเคลื่อนจากขดลวดอันใดอันหนึ่ง เมื่อเซนเซอร์ได้รับการเปลี่ยนแปลงความดันไดอะแฟรม จะเลื่อนจากขดลวดหนึ่งไปยังอีกอันหนึ่ง ไดอะแฟรมเป็นวัสดุที่มีคุณสมบัติทางด้านสนามแม่เหล็ก เมื่ออยู่ใกล้ขดลวดใดจะทำให้ความหนาแน่นของฟลักซ์แม่เหล็กรอบๆขดลวดสูงขึ้น เมื่อสนามแม่เหล็กรอบขดลวดสูงขึ้นจะส่งผลให้เพิ่มความสามารถในการนำไฟฟ้า ในขณะที่เดียวกันขดลวดอีกอัน ก็ลดความต้านทานเป็นผลให้ขดลวดทั้ง 2 มีคุณสมบัติไม่สมดุลกัน ส่งผลให้มีสัญญาณขนาดเล็ก ส่งมาจากสายสัญญาณ

การเปลี่ยนแปลงของความต้านทานของขดลวด มีผลเป็นสัดส่วนกับการเปลี่ยนแปลงตำแหน่งของไดอะแฟรม ดังนั้นแอมพลิจูดของสัญญาณ ซึ่งเป็นผลที่ได้จากการเปลี่ยนแปลงความ

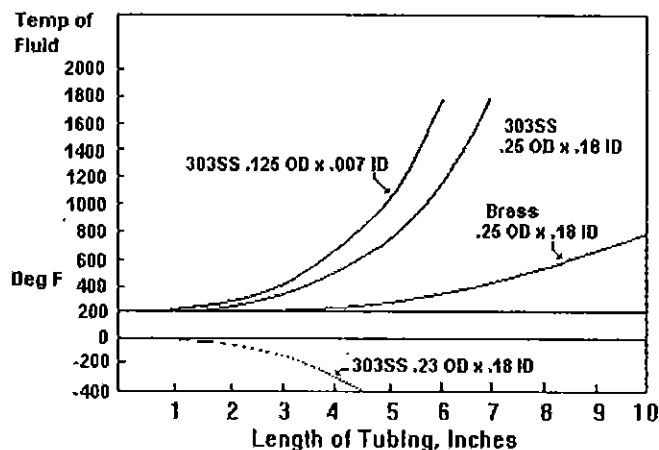
คัมพเลกของสัญญาณ จะบอกให้รู้ถึงทิศทางการเคลื่อนที่ของไดอะแฟรม ดังนั้นแอมพลิจูดของสัญญาณ จึงเป็นผลที่ได้จากการเปลี่ยนแปลงความคัมพเลกของสัญญาณ จะบอกให้รู้ถึงทิศทางการเคลื่อนที่ของไดอะแฟรม เอาท์พุทที่ได้จากวงจรวัดการเปลี่ยนแปลงมีขนาดเต็มสเกลได้ 20 มิลลิโวลต์หรืออาจมากกว่า

เนื่องจากเอาท์พุทที่ได้ขึ้นอยู่กับารเคลื่อนที่ของไดอะแฟรมสเกลของค่าความคัมพเลกที่วัดได้ จะขึ้นอยู่กับความกว้างของพื้นที่ที่ไดอะแฟรมเคลื่อนที่

ในบางครั้งจำเป็นต้องวัดแรงดันของไหลที่อุณหภูมิ ซึ่งต่างไปจากปกติ อาจมากหรือน้อยกว่าปกติ เราหลีกเลี่ยงวิธีการที่จะสร้างอุปกรณ์ตรวจวัดความคัมพเลกพิเศษที่ทนต่ออุณหภูมิได้โดยการแยกเซนเซอร์วัดความคัมพเลกออกจากจุดที่ต้องการวัด โดยให้แรงดันผ่านหลอดหรือท่อสั้นๆ ดังรูปที่ 2.3



รูปที่ 2.3 วัดความคัมพเลกของไหลจากอุณหภูมิ [4]



รูปที่ 2.4 กราฟแสดงให้เห็นถึงความสัมพันธ์ของขนาดความยาวของท่อกับอุณหภูมิ [4]

จากกราฟดังรูปที่ 2.4 แสดงให้เห็นถึงความสัมพันธ์ของขนาดความยาวของท่อกับอุณหภูมิ เมื่อต้องการวัดความดันที่อุณหภูมิต่างกัน

ในการวัดความดันจะไม่มีของไหลผ่านท่อมายังเครื่องมือและเนื่องจากการควบคุมอุณหภูมิที่ปลายท่อเครื่องมือวัด จึงไม่ได้รับผลกระทบจากอุณหภูมिनอกจากว่าของไหลที่จะวัดมีอุณหภูมิสูงมากเกินไป

2.5 ไมโครคอนโทรลเลอร์

ในโครงการนี้ใช้ ไมโครคอนโทรลเลอร์ PIC 16F877 [5] คือไมโครคอนโทรลเลอร์อีกตระกูลหนึ่ง ซึ่งแนวคิดของไมโครคอนโทรลเลอร์ตระกูลนี้ก็คือ พยายามรวมเอาทุกอย่างเอาไว้ในตัว โดยไม่จำเป็นต้องต่ออุปกรณ์เสริมจากภายนอก ในตัวของไมโครคอนโทรลเลอร์รุ่นนี้ จะมีฟังก์ชันที่ใช้ในการประมวลผลรวมทั้งหน่วยความจำ ซึ่งทำให้เหมือนกับซีพียูตัวหนึ่ง

2.5.1 ความเร็วของไมโครคอนโทรลเลอร์ PIC 16F877

ภาคของความถี่สัญญาณนาฬิกา สามารถทำสัญญาณนาฬิกาได้ที่ 20 เมกะเฮิร์ตซ์ ซึ่งทำให้หนึ่งคำสั่งของไมโครคอนโทรลเลอร์ PIC 16F877 ใช้เวลาเพียง 0.25 ไมโครวินาที แต่อย่างไรก็ตาม ได้มีบริษัทอื่นชื่อลิจิทิวิตี ไมโครคอนโทรลเลอร์ PIC 16F877 จากไมโครชิพและได้สร้างชิพที่มีความเร็วได้มากกว่าเดิมขึ้นไปอีก

2.5.2 หน่วยความจำของไมโครคอนโทรลเลอร์ PIC 16F877

ในอดีตนหน่วยความจำของไมโครคอนโทรลเลอร์ PIC 16F877 จะค่อนข้างน้อย คืออยู่ระหว่าง 512 เวิร์ดถึง 4 กิโลเวิร์ดแต่ในปัจจุบันบริษัทไมโครชิพซึ่งเป็นเจ้าของไมโครคอนโทรลเลอร์ PIC 16F877 ได้พัฒนาจนทำให้หน่วยความจำของไมโครคอนโทรลเลอร์ PIC 16F877 มีขนาดเป็นหลายสิบกิโลไบต์ และมีที่ท่าว่าจะขยายได้ใหญ่ขึ้นเรื่อยๆ ในเรื่องของความนับขนาดของหน่วยความจำของไมโครคอนโทรลเลอร์ PIC 16F877 จะนับไม่เหมือนปกติ โดยที่หนึ่งคำสั่งของไมโครคอนโทรลเลอร์ PIC 16F877 จะมีขนาด 14 บิตดังนั้นเราจะเรียกว่า 1 เวิร์ดของไมโครคอนโทรลเลอร์ PIC 16F877 จะมีขนาด 14 บิต เช่น PIC16F84A ระบุว่ามีความจำ 1 กิโล (ซึ่งหมายถึง 1 กิโลเวิร์ดถ้าคำนวณให้เป็นแบบ 1 ไบต์ = 8 บิต จะได้ว่า $1 \times 1,024 \times 14 = 14,336$ บิต ดังนั้นก็คือ $14,336 / (8 \times 1,024) = 1.75$ กิโลไบต์

2.5.3 คุณสมบัติของไมโครคอนโทรลเลอร์ PIC 16F877

คุณสมบัติต่างๆของไมโครคอนโทรลเลอร์ที่เราควรทราบ เช่น ความถี่สูงสุดในการทำงาน, ค่าของหน่วยความจำประเภทต่างๆ, แรงดันในการทำงาน, จำนวนพอร์ต I/O, ตอบสนองการอินเตอร์รัพท์ได้ที่แห่ง, มีไทม์เมอร์ที่ตัว, มีพีคบัปเบิลยูดีทั้งหมดที่ชุด เป็นต้น โดยมีคุณสมบัติของไมโครคอนโทรลเลอร์ PIC16F877 มีดังนี้คือ

- มีคำสั่งในภาษาแอสเซมบลี 35 คำสั่ง
- ใช้ความถี่ออสซิลเลเตอร์ได้สูงสุด 20 เมกกะเฮิร์ตซ์
- มีหน่วยความจำโปรแกรม แฟลชเมมโมรี่ขนาด 8 กิโลเวิร์ด (14-บิตเวิร์ด)
- มีหน่วยความจำข้อมูลแบบแรม 368 ไบต์
- มีหน่วยความจำข้อมูลแบบ อีอีพรอม 256 ไบต์
- มีการตอบสนองอินเตอร์รัพท์ทั้งหมด 14 แหล่ง
- สามารถเลือกระดับการป้องกันข้อมูล (Code Protection) ได้
- มีโหมดประหยัดพลังงาน (Sleep Mode)
- สามารถเลือกแหล่งสัญญาณนาฬิกาหลายโหมด XT RC และออสซิลเลเตอร์พลังงานต่ำ
- มีฟังก์ชันการรักษาเสถียรภาพการทำงาน ได้แก่ POR, PWRT, OST, BOR และ WDT
- การโปรแกรมตัวชิพแบบ ICSP (ICSP: In-Circuit Serial Programming)

- สามารถทำงานที่ไฟเลี้ยงวงจรตั้งแต่ 2.0 โวลต์ถึง 5.5 โวลต์
- ขาพอร์ต I/O แต่ละขา สามารถรับและปล่อยกระแสได้สูงสุด 25 มิลลิแอมป์
- มีโมดูล ไทม์เมอร์/เคาท์เตอร์ ใช้งานทั้งหมด 3 ตัว ไทม์เมอร์ 0, ไทม์เมอร์ 1, และ ไทม์เมอร์ 2
- มีโมดูล CCP (CCP: Compare / Capture / PWM) จำนวน 2 ชุด
- มีโมดูล อนุาล็อก ทู คิจิตอล คอนเวอร์เตอร์ (Analog to Digital Con) ความละเอียด ขนาด 8 บิต และ 10 บิต จำนวน 8 ช่อง

2.5.4 วงจรภายในไมโครคอนโทรลเลอร์

- มีโมดูลสื่อสารอนุกรมแบบ USART (USART: Universal Synchronous Asynchronous Receiver / Transmitter)

- มีพอร์ต I/O จำนวน 5 พอร์ต ได้แก่ พอร์ต A, B, C, D และ E มีขา I/O รวมกัน 33 ขา
 ดังนี้คือ

พอร์ต A มี 6 ขา RA0 ถึง RA5

พอร์ต B มี 8 ขา RB0 ถึง RB7

พอร์ต C มี 8 ขา RC0 ถึง RC7

พอร์ต D มี 8 ขา RD0 ถึง RD7

พอร์ต E มี 3 ขา RE0 ถึง RE2

2.6 โปรแกรมไมโครซอฟท์วิซวลเบสิก (Microsoft Visual Basic)

วิซวลเบสิก (Visual Basic) หรือ VB เป็นโปรแกรมภาษาคอมพิวเตอร์ตัวหนึ่ง ซึ่งใช้สำหรับสร้างหรือพัฒนาโปรแกรมใช้งานบนวินโดว มีความสามารถในการทำงานที่คล้ายกับภาษาคอมพิวเตอร์อื่นๆ เช่น C, Pascal, C++, C# สร้างโดยบริษัทไมโครซอฟท์ ภาษานี้เป็นหนึ่งในภาษาโปรแกรมยอดนิยมสำหรับโปรแกรมที่ใช้ในด้านธุรกิจ ซึ่งเป็นภาษาคอมพิวเตอร์ระดับสูง ใช้งานง่าย เหมาะสำหรับผู้เริ่มต้น เพราะใช้คำในภาษาอังกฤษที่เข้าใจง่าย และเมื่อเป็นวิซวลเบสิกซึ่งใช้ลักษณะของการมองเห็นได้ วิซวลที่เป็นการติดต่อกับผู้ใช้ด้วยกราฟิกหรือรูปภาพ (Graphical User Interface-GUI) จึงทำให้การพัฒนาโปรแกรมใช้งานได้สะดวกและรวดเร็วขึ้น ถึงแม้จะใช้งาน

ง่าย แต่ก็มีขีดความสามารถสูง เหมาะสำหรับการพัฒนาโปรแกรมใช้งานได้หลายด้าน เช่น งานคำนวณทั่วไป งานด้านฐานข้อมูล เกม ฯลฯ และยังได้พัฒนาต่อเป็นภาษา (VB.NET) อีกด้วย

ความหมายของคอทเน็ต (.NET) เป็นกลุ่มของเทคโนโลยีทางซอฟต์แวร์ที่เชื่อมโยงข้อมูล, ข่าวสาร, คน และระบบและอุปกรณ์ต่างๆ เข้าด้วยกัน ด้านการเขียนโปรแกรมคอทเน็ต หมายถึง เค้าโครง (Framework) คือ สภาพแวดล้อมที่สนับสนุนการพัฒนาและรัน โปรแกรมในรูปแบบของเค้าโครงคอทเน็ต จะทำหน้าที่ควบคุมการรัน โปรแกรม และให้บริการทรัพยากรต่างๆ แก่โปรแกรมที่รัน เช่น การโหลดโปรแกรมขึ้นมาทำงาน การจัดการหน่วยความจำ การจัดเตรียมไลบรารีให้โปรแกรมเรียกใช้งาน

2.6.1 เริ่มต้นใช้งานโปรแกรม (Microsoft Visual Basic)

การเปิดใช้งานโปรแกรม สามารถทำได้ดังนี้

1. คลิกที่ปุ่มเริ่มทำงาน (Start)
2. เลือก All Program
3. เลือก Microsoft Visual Studio
4. เลือก Microsoft Visual Basic
5. รอสักครู่จะปรากฏหน้าจอโปรแกรมขึ้นมา

- แถบหัวเรื่อง (Title bar) : เป็นแถบแสดงหัวเรื่องโดยปกติจะบอกชื่อของโปรแกรม

- แถบเมนู (Menu bar) : เป็นแถบแสดงรายการคำสั่งต่างๆ ของโปรแกรม

- แถบเครื่องมือ (Tool bar) : เป็นแถบเครื่องมือที่รวบรวมไอคอนต่างๆ ที่ใช้ประจำ

หน้าต่างเริ่มต้น (Start Page) : เป็นหน้าต่างเริ่มการทำงานของโปรแกรม ซึ่งจะมีส่วนที่แสดงรายชื่อโปรเจก (Project) ที่เราได้เคยสร้างไว้แล้วในกรณีที่เพิ่งเปิดใช้งานครั้งแรกจะไม่มีรายชื่อปรากฏอยู่

- กล่องเครื่องมือ (Toolbox) : เป็นส่วนที่เก็บเครื่องมือต่างๆ ที่เราจะนำมาใช้งานในโปรเจกของเรา แต่ในกรณียังไม่มีเครื่องมือเนื่องจากเรายังไม่ได้สร้างโปรเจกใดๆ

- หน้าต่าง (Solution Explorer) : หน้าต่างที่แสดงรายชื่อและจัดการกับโปรเจกของเรา

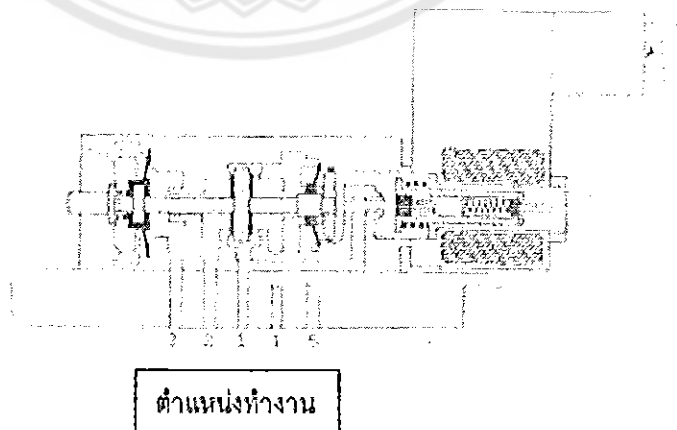
- แถบสถานะ (Status Bar) : เป็นแถบที่มีไว้สำหรับบอกสถานะการทำงานของโปรแกรม

2.7 โซลินอยด์วาล์ว

โซลินอยด์ (Solenoid) [7] เป็นอุปกรณ์แม่เหล็กไฟฟ้าชนิดหนึ่งที่มีหลักการทำงานคล้ายกับรีเลย์ (Relay) ภายในโครงสร้างของโซลินอยด์จะประกอบด้วยขดลวดที่พันอยู่รอบแท่งเหล็กที่อยู่ในประกอบด้วยแม่เหล็กชุดบนกับชุดล่าง เมื่อมีกระแสไฟฟ้าไหลผ่านขดลวดที่พันรอบแท่งเหล็ก ทำให้แท่งเหล็กชุดล่างมีอำนาจแม่เหล็กดึงแท่งเหล็กชุดบนลงมาสัมผัสกัน ทำให้ครบวงจรทำงาน เมื่อวงจรถูกตัดกระแสไฟฟ้าทำให้แท่งเหล็กส่วนล่างหมดอำนาจแม่เหล็ก สปริงก็จะดันแท่งเหล็กส่วนบนกลับสู่ตำแหน่งปกติ จากหลักการดังกล่าวของโซลินอยด์ก็จะนำมาใช้ในการเคลื่อนลิ้นวาล์วของระบบนิวแมติกส์ไฟฟ้า โครงสร้างของโซลินอยด์วาล์ว โดยทั่วไปแบ่งออกเป็น 2 ชนิดคือ เลื่อนวาล์วกลับด้วยแรงสปริง (Single Solenoid Valve) และเลื่อนวาล์วกลับด้วยโซลินอยด์วาล์ว (Double Solenoid Valve)

2.7.1 วาล์วปกติปิด เลื่อนวาล์วกลับด้วยแรงสปริง (Way Single Solenoid Valve)

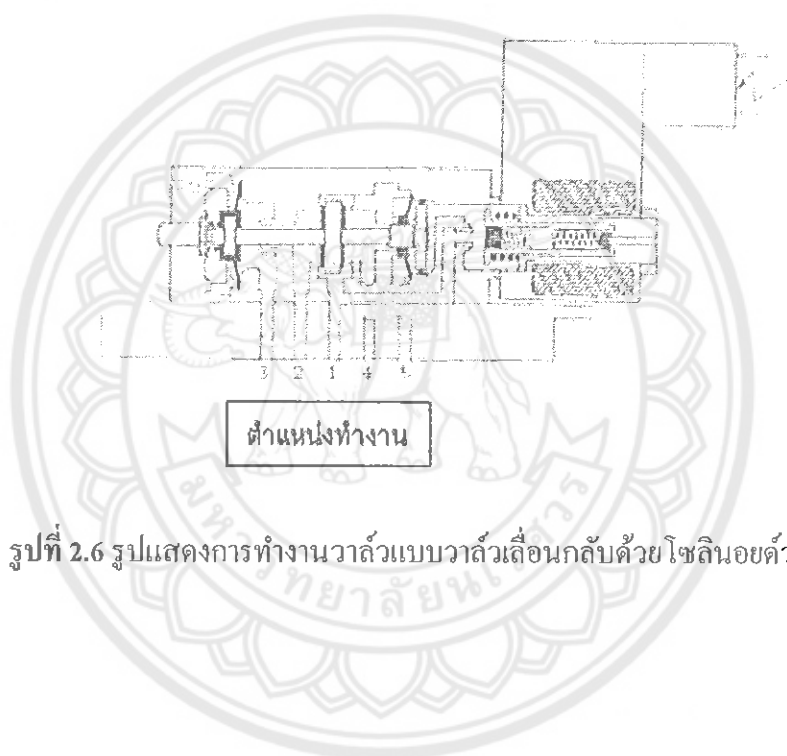
หลักการทำงานตำแหน่งปกติ โซลินอยด์วาล์วยังไม่มีกระแสไฟฟ้าไหลผ่าน ทำให้ไม่มีอำนาจแม่เหล็ก สปริงจะดันลิ้นวาล์วให้ปิด ตำแหน่งทำงานเมื่อกดปุ่มเริ่ม ทำให้มีกระแสไฟฟ้าไหลผ่าน โซลินอยด์วาล์ว ทำให้เกิดอำนาจแม่เหล็กไปเคลื่อนลิ้นวาล์ว เมื่อปล่อยปุ่มเริ่มจะทำให้โซลินอยด์วาล์วหมดอำนาจแม่เหล็ก สปริงจะดันลิ้นวาล์วให้กลับสู่สถานะปกติ โดยมีตำแหน่งการทำงานแสดงดังรูปที่ 2.5



รูปที่ 2.5 รูปแสดงการทำงานวาล์วแบบเลื่อนวาล์วกลับด้วยแรงสปริง [7]

2.7.2 วาล์วเลื่อนกลับด้วยโซลินอยด์วาล์ว (Way Double Solenoid Valve)

หลักการการทำงานตำแหน่งปกติโซลินอยด์วาล์วยังไม่มีกระแสไฟฟ้าไหลผ่าน ทำให้ไม่มีอำนาจแม่เหล็ก เมื่ออยู่ในสภาวะปกติ ตำแหน่งทำงานเมื่อกดปุ่มเริ่ม ทำให้มีกระแสไฟฟ้าไหลผ่าน โซลินอยด์วาล์ว ทำให้เกิดอำนาจแม่เหล็กไปเลื่อนลิ้นวาล์ว เมื่อปล่อยปุ่มเริ่มที่หนึ่ง จะทำให้โซลินอยด์วาล์วตัวที่หนึ่งค้างสภาวะเดิมอยู่ เมื่อกดปุ่มเริ่มที่สองทำให้กระแสไฟฟ้าไหลผ่าน โซลินอยด์วาล์วตัวที่สอง ทำให้เกิดอำนาจแม่เหล็กเลื่อนลิ้นวาล์ว เมื่อปล่อยปุ่มเริ่มตัวที่สองจะทำให้โซลินอยด์วาล์ว ตัวที่สองค้างอยู่สภาวะเดิมเช่นเดียวกัน โดยมีตำแหน่งการทำงานแสดงดังรูปที่ 2.6

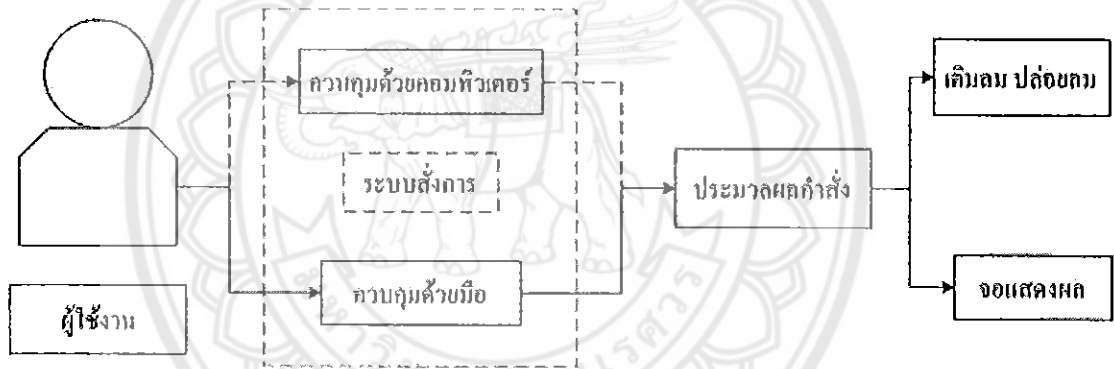


รูปที่ 2.6 รูปแสดงการทำงานวาล์วแบบวาล์วเลื่อนกลับด้วย โซลินอยด์วาล์ว [7]

บทที่ 3

ขั้นตอนการดำเนินงาน

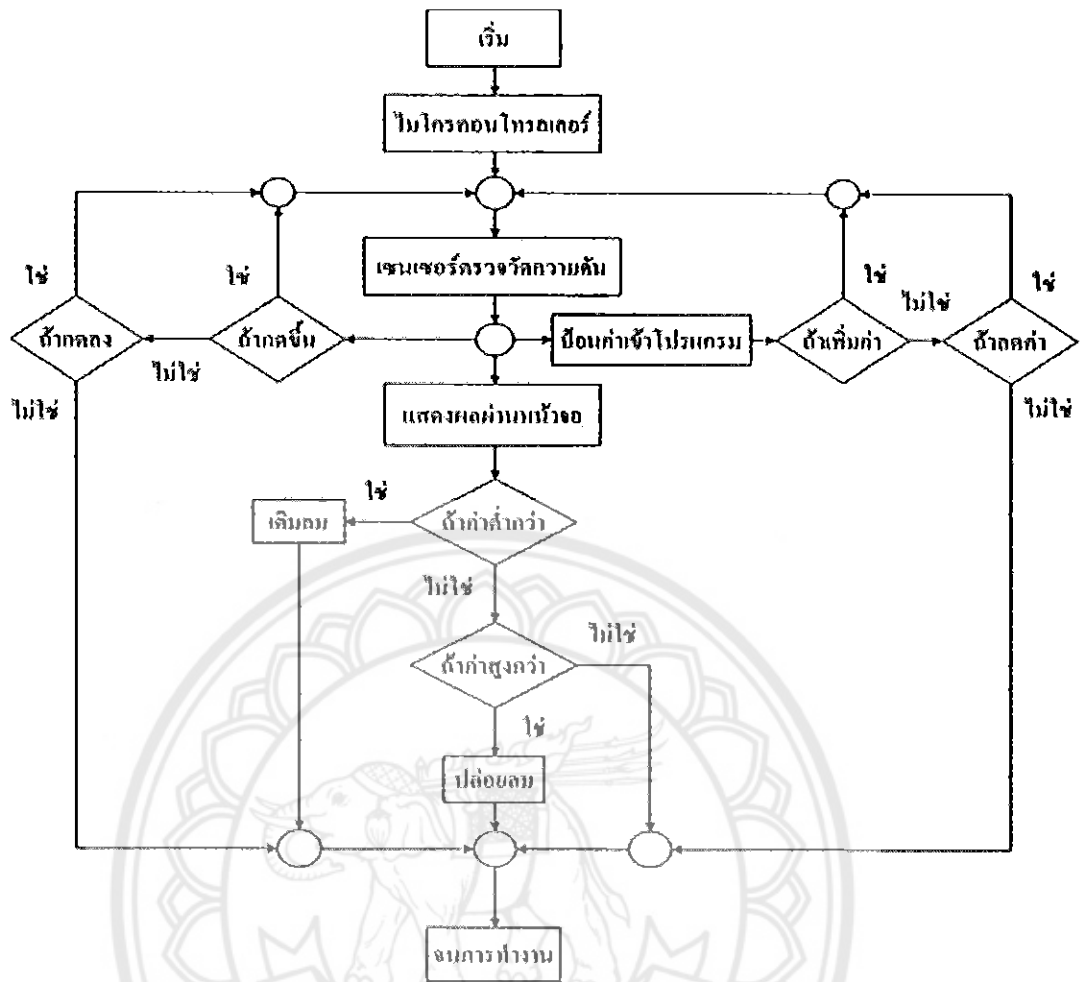
เครื่องเติมลมอัตโนมัติควบคุมโดยไมโครคอนโทรลเลอร์ในกรณีศึกษา นี้ เป็นการควบคุมด้วยไมโครคอนโทรลเลอร์ PIC 16F877 ผ่านคอมพิวเตอร์ ซึ่งหลังจากได้ศึกษาทฤษฎีที่มีความเกี่ยวข้อง หลักการทำงานของอุปกรณ์ที่ต้องการนำมาใช้งาน เราจึงทำการออกแบบและจัดหาอุปกรณ์ในการทำโครงงานนี้ โดยมีการแบ่งขั้นตอนการทำงานดังแสดงดังรูปที่ 3.1 แยกการทำงานออกเป็น 4 ส่วน คือ ส่วนระบบสั่งการ ส่วนประมวลผลคำสั่ง ส่วนเติมลม ปลัวยลม และระบบจอแสดงผลการทำงาน



รูปที่ 3.1 แผนภาพแสดงภาพรวมของระบบ (System Overview Diagram)

3.1 ระบบสั่งการ

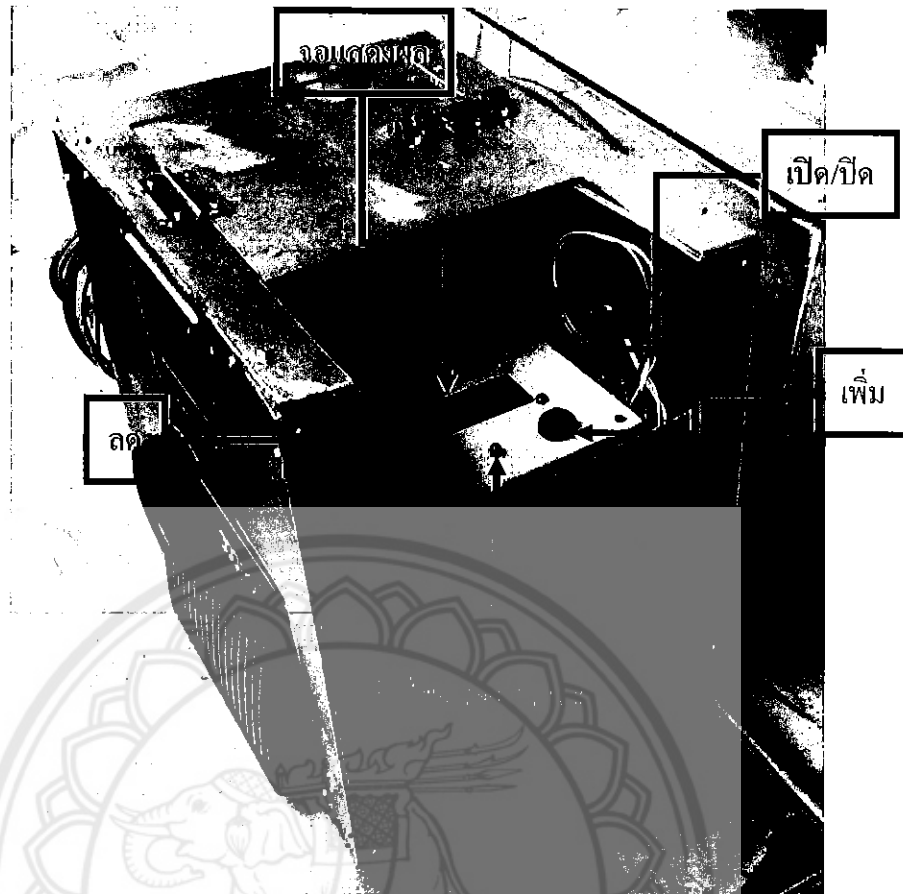
ในการควบคุมการทำงานของเครื่องเติมลมอัตโนมัติการใช้งานสามารถเลือกควบคุมได้ 2 วิธี คือ ระบบสั่งการควบคุมด้วยมือและระบบสั่งการด้วยคอมพิวเตอร์ โดยมีหลักการทำงานดังนี้ เมื่อเริ่มทำงานไมโครคอนโทรลเลอร์จะสั่งให้เซนเซอร์วัดความดันภายในล้อรถยนต์ และถ้าทำการกดปุ่มเพิ่มหรือเพิ่มค่าผ่าน โปรแกรมวิซวลเบสิก จากนั้นทำการเปรียบเทียบค่า ถ้าค่าความดันที่วัดต่ำกว่าค่าที่ป้อนจากการกด ปุ่มลมก็จะทำการเติมลมจนค่าความดันทั้งสองเท่ากัน เครื่องก็จะหยุดทำงาน และถ้าทำการกดปุ่มลดหรือลดค่าผ่าน โปรแกรมวิซวลเบสิก จากนั้นทำการเปรียบเทียบค่า ถ้าค่าความดันที่วัดสูงกว่าค่าที่ป้อน ปุ่มลมก็จะทำการปล่อยลมจนค่าความดันทั้งสองเท่ากัน เครื่องก็จะหยุดทำงาน แสดงดังรูปที่ 3.2



รูปที่ 3.2 การทำงานของเครื่องเติมลมอัตโนมัติ

3.1.1 ระบบสั่งการควบคุมด้วยมือ (MANUAL)

การควบคุมการทำงานของเครื่องเติมนั้น การใช้โหมดควบคุมด้วยมือเพื่อใช้เป็นคำสั่งให้กับส่วนประมวลผลโดยตรง ทำให้การทำงานของเครื่องเติมนสะดวกรู้สึกและกรณีที่ไม่ได้ใช้งานคอมพิวเตอร์ ผู้ใช้ยังสามารถควบคุมเครื่องเติมลมได้ ลักษณะปุ่มกดรับคำสั่งจะอยู่ด้านหน้าเครื่องเติมลมแสดงดังรูปที่ 3.3



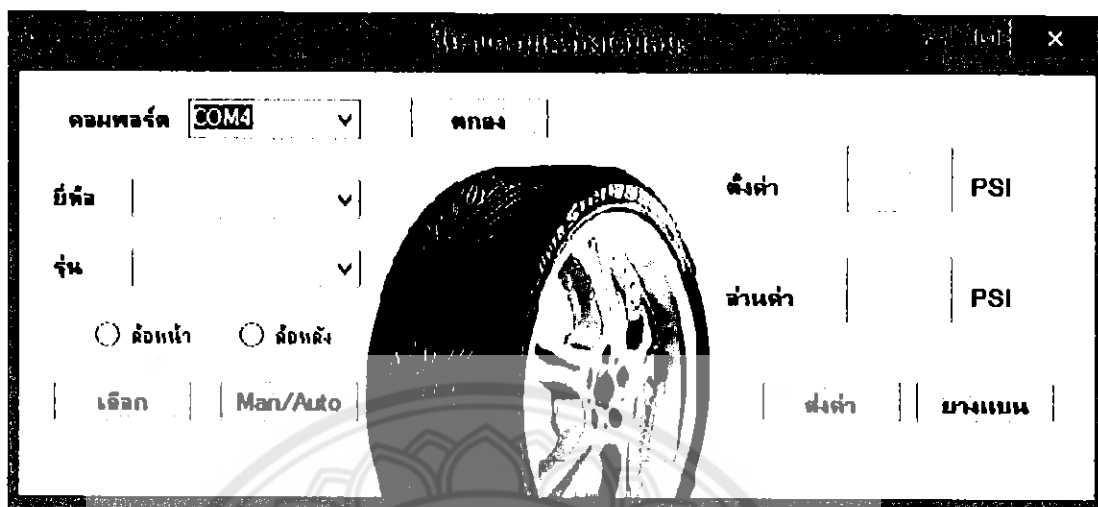
รูปที่ 3.3 แสดงปุ่มรับคำสั่งการควบคุมด้วยมือ

จากรูปที่ 3.3 เมื่อเปิดสวิตช์ จอนัดคงผลจะแสดงค่าแรงดันสองค่าคือแรงดันที่วัดได้จากขางรถยนต์และแรงดันที่เรากำหนดไว้ ซึ่งกำหนดไว้ที่ 35 ปอนด์ต่อตารางนิ้ว โดยที่เครื่องจะทำการเปรียบเทียบค่า ถ้าแรงดันที่วัดได้มีค่าต่ำกว่าแรงดันที่กำหนด เครื่องก็จะทำการเติมลมทันที ถ้าแรงดันที่วัดได้มีค่าสูงกว่าแรงดันที่กำหนด เครื่องก็จะทำการปล่อยลม และเมื่อแรงดันที่วัดได้เท่ากับแรงดันที่กำหนด เครื่องจะหยุดทำงานไฟแสดงสถานะจะกระพริบ สามารถเพิ่มลดแรงดันได้โดยการกดปุ่ม

3.1.2 ระบบสั่งการด้วยคอมพิวเตอร์

ในการควบคุมการทำงานของเครื่องเติมลมอัตโนมัติ นั้น จะใช้หน้าต่างรับคำสั่งโปรแกรมที่สร้างขึ้นจาก โปรแกรมวิซวลเบสิก (Visual Basic) โดยที่ผู้ใช้สามารถเลือกสถานะในการทำงานได้จาก โปรแกรมรับคำสั่งจากผู้ใช้งานทางคอมพิวเตอร์ที่สร้างขึ้น เพื่อใช้ในการสื่อสารระหว่างพอร์ตอนุกรมของคอมพิวเตอร์กับ ไมโครคอนโทรลเลอร์ โดยใช้ชุดแปลงสัญญาณพอร์ตอนุกรม RS-232 เป็นตัวกลางส่งคำสั่งควบคุมไปยังเครื่องเติมลมอัตโนมัติ ในการสื่อสารระหว่างพอร์ตอนุกรม

RS-232 ความเร็วของการรับส่งข้อมูลสำหรับใน โครงงานนี้ใช้เป็น 9600 บิตต่อวินาที โดยหน้าต่างโปรแกรมแสดงได้ดังรูปที่ 3.4

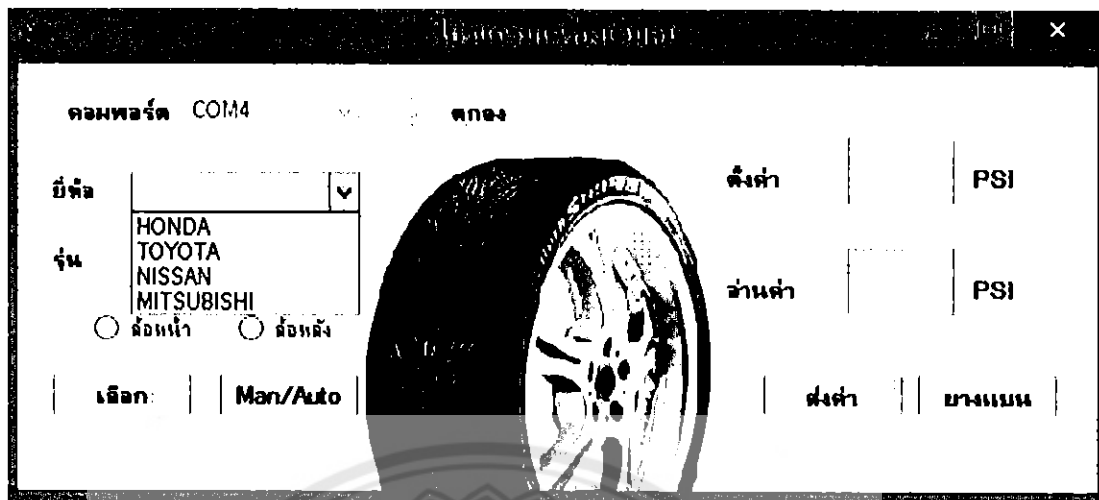


รูปที่ 3.4 รูปแสดงปุ่มรับคำสั่งการควบคุมด้วยคอมพิวเตอร์

จากรูปที่ 3.4 ลักษณะการทำงานคือ เมื่อเปิดโปรแกรมขึ้นมาตัวโปรแกรมจะทำการเชื่อมต่อกับคอมพอร์ตอัตโนมัติเราต้องกดตกลงเพื่อยืนยันคอมพอร์ต โดยตัวโปรแกรมสามารถทำงานได้สองลักษณะคือ แบบอัตโนมัติและแบบป้อนค่า โดยกดเลือกที่ปุ่ม (Man/Auto) หลังจากนั้นเราก็จะสามารถเติมลมได้ตามแบบที่เลือก

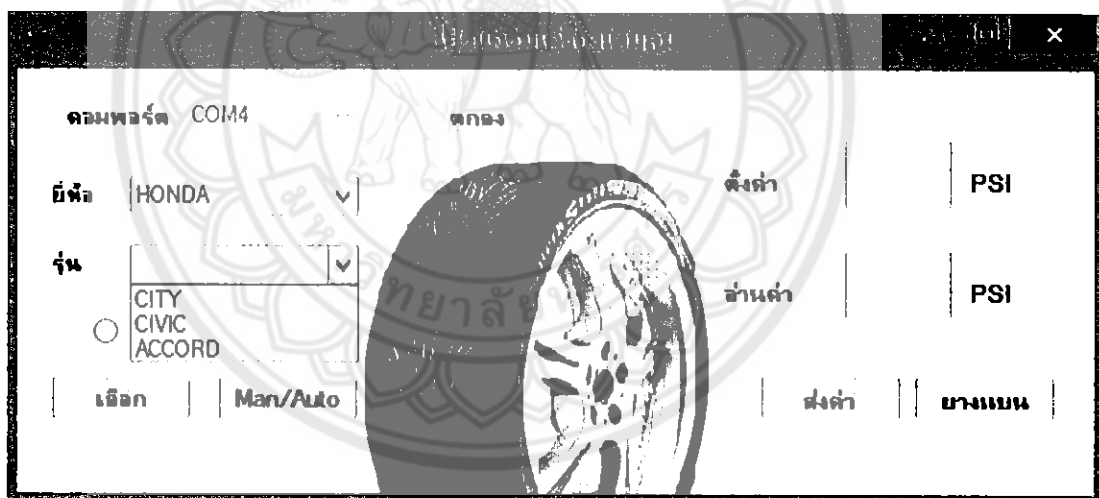
1. แบบอัตโนมัติ ในแบบนี้เราสามารถเลือกยี่ห้อรถ เลือกรุ่น และเลือกล้อได้ เมื่อทำการเลือกยี่ห้อรถ เลือกรุ่น และเลือกล้อ ให้ทำการกดเลือก ค่าของแรงดันก็จะปรากฏในช่องตั้งค่าทันที จากนั้นทำการกดส่งค่า เครื่องก็จะทำการเติมลมจนกว่าแรงดันที่อ่านค่าได้จะเท่ากับค่าที่เรากำหนด เครื่องก็จะหยุดการทำงาน ซึ่งมีขั้นตอนการทำงานของโปรแกรมแสดงดังรูปที่ 3.5-3.8

หลังจากเลือกคอมพิวเตอร์แล้ว ให้ทำการเลือกยี่ห้อ โดยแสดงดังรูปที่ 3.5



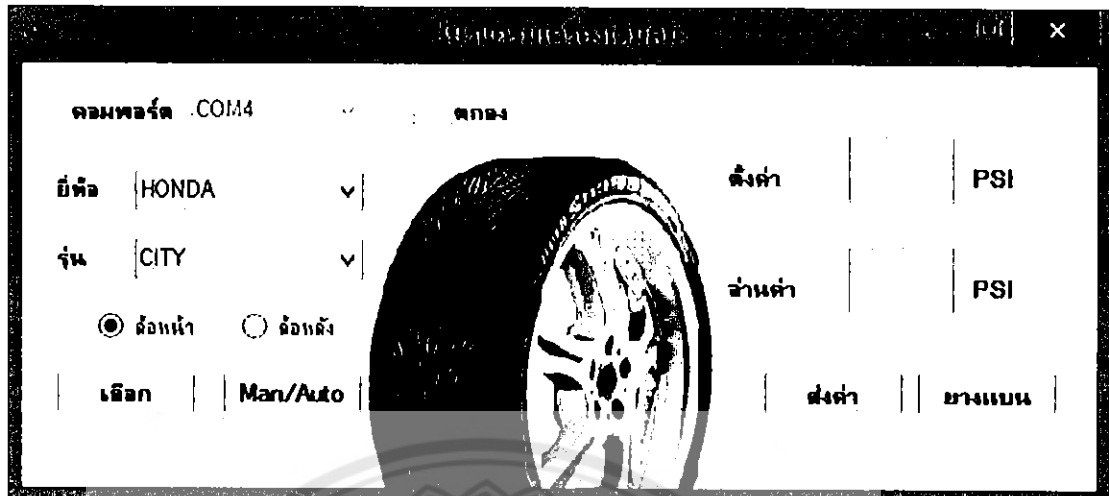
รูปที่ 3.5 รูปแสดงการเลือกยี่ห้อ

เมื่อทำการเลือกยี่ห้อเสร็จ ให้ทำการเลือกรุ่น โดยแสดงดังรูปที่ 3.6



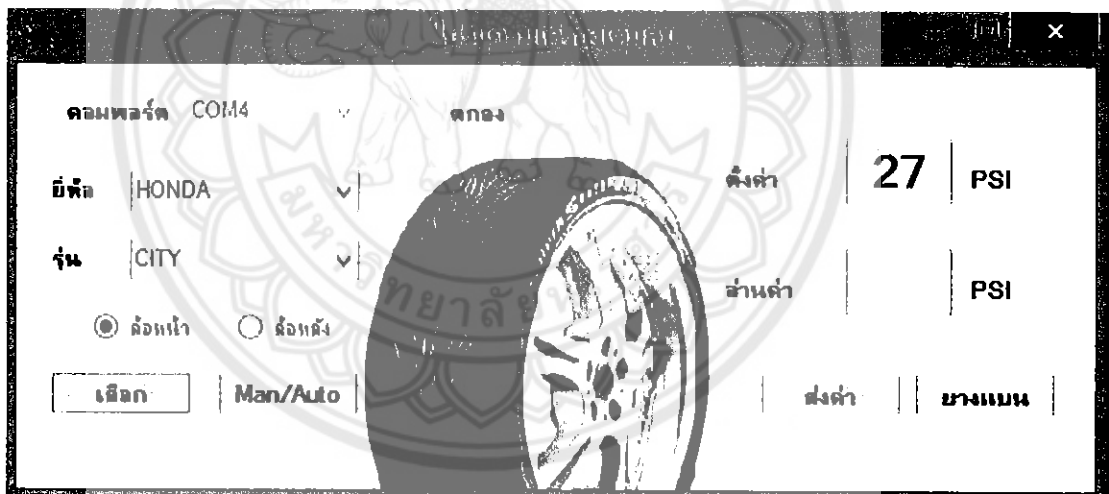
รูปที่ 3.6 รูปแสดงการเลือกรุ่น

และเมื่อเลือกรุ่นเรียบร้อยแล้ว ให้ล้อที่ท่านต้องการเติม โดยแสดงดังรูปที่ 3.7



รูปที่ 3.7 รูปแสดงการเลือกล้อ

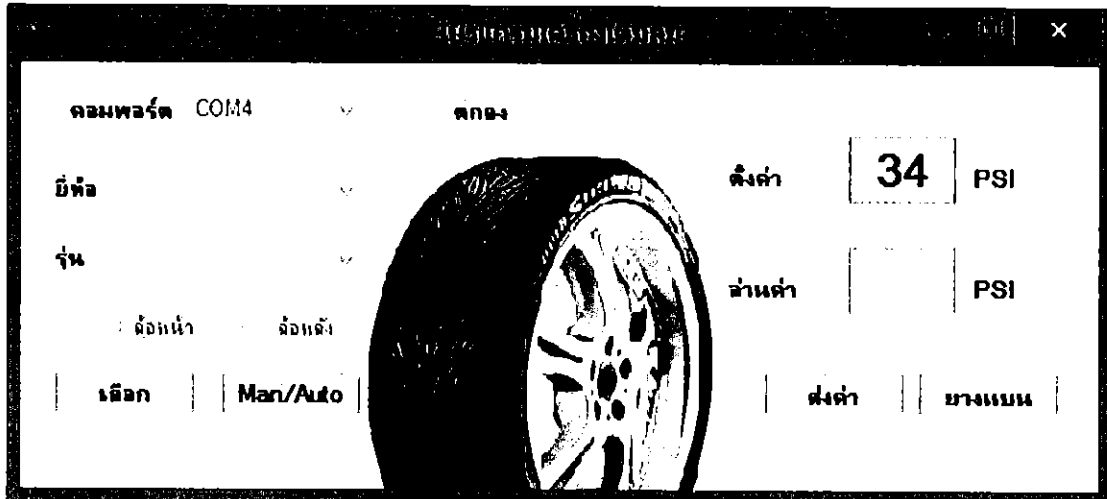
จากนั้นทำการกดปุ่มเลือก ค่าแรงดันจะแสดงขึ้นที่ช่องตั้งค่า โดยแสดงดังรูปที่ 3.8



รูปที่ 3.8 รูปแสดงระบบสั่งการผ่านคอมพิวเตอร์แบบอัตโนมัติ

จากรูปที่ 3.8 เมื่อทำตามขั้นตอนที่ได้กล่าวไว้เบื้องต้นเสร็จเรียบร้อยแล้ว ให้ทำการกดส่งค่า เครื่องก็จะทำการเติมตามแรงดันที่กำหนด

2. แบบป้อนค่า ในแบบนี้หากรุ่นของรถยนต์ที่ต้องการเติมไม่มีในแบบอัตโนมัติ เราสามารถทำการป้อนค่าแรงดัน โดยทำการกดปุ่ม “Man/Auto” หลังจากนั้นจะสามารถเติมค่าในช่องตั้งค่าได้ โดยแสดงดังรูปที่ 3.9



รูปที่ 3.9 รูปแสดงระบบสั่งการผ่านคอมพิวเตอร์แบบป้อนค่า

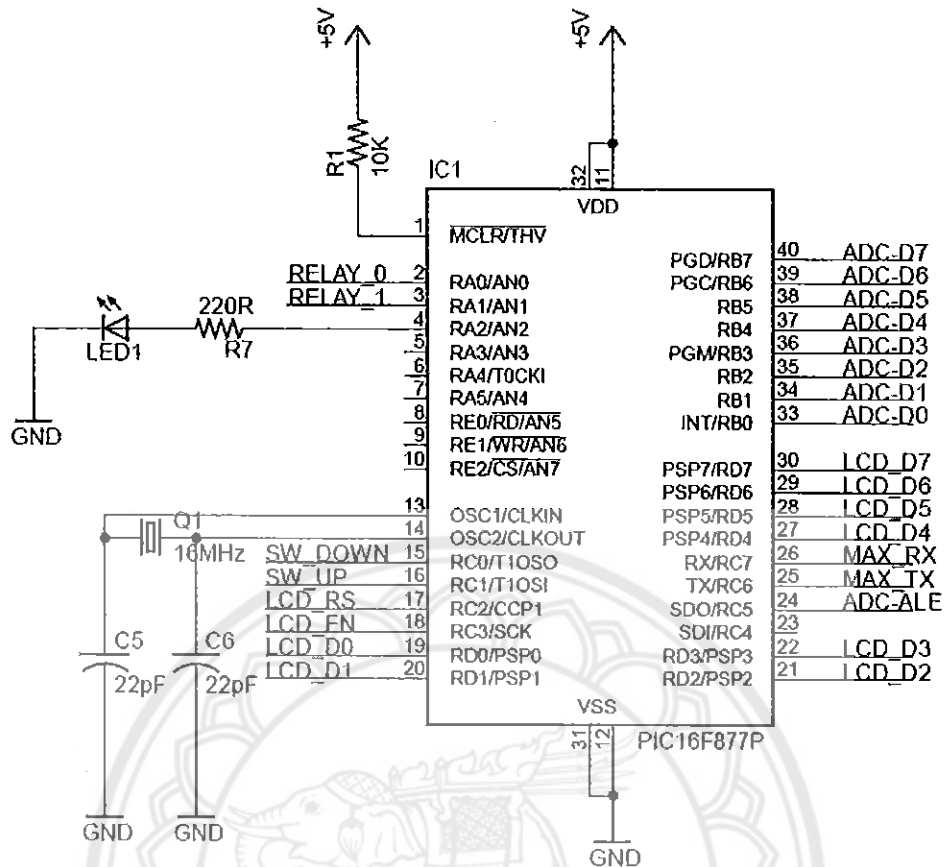
จากรูปที่ 3.9 เมื่อทำการป้อนค่าเสร็จ ให้ทำการกดปุ่มส่งค่า เครื่องเติมลมก็จะทำการเติมลมตามแรงดันที่กำหนด

หมายเหตุ : ปุ่มยางแบนมีไว้สำหรับในกรณีที่ยางรถยนต์ไม่มีแรงดันอยู่เลยหรือมีค่าเท่ากับ 0 ปอนด์ต่อตารางนิ้ว

3.2 ระบบประมวลผลคำสั่ง

3.2.1 วงจรไมโครคอนโทรลเลอร์

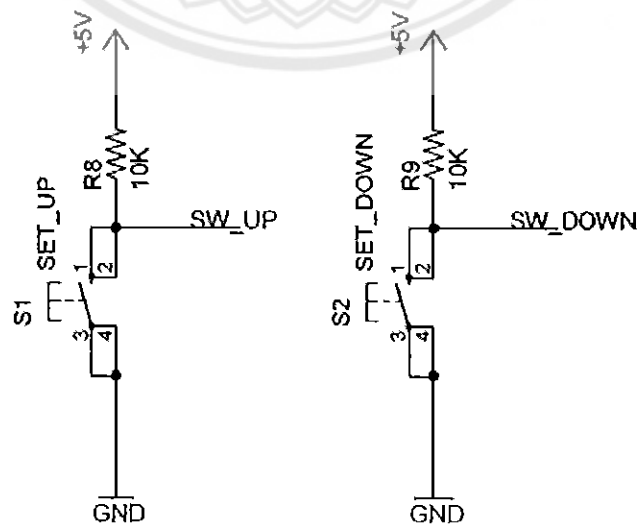
วงจรไมโครคอนโทรลเลอร์ควบคุมการทำงานทั้งหมดของระบบ โดยมีการเชื่อมต่อกับอุปกรณ์ที่ใช้ในการควบคุมคืออินพุตประกอบด้วย วงจรสวิตช์ควบคุมแรงดัน วงจรเซนเซอร์ตรวจวัดแรงดัน วงจรควบคุมโซลีนอยด์วาล์ว และเอาต์พุตประกอบด้วยวงจรควบคุมเครื่องปั๊มลม หน้าจอแสดงผลแบบดิจิทัล ซึ่งรับไฟฟ้าจากแบตเตอรี่ 12 โวลต์ โดยผ่าน MAX232 ที่ทำหน้าที่ในการแปลงจากแรงดันสูงให้เหลือ 0 – 5 โวลต์ โดยแสดงดังรูปที่ 3.10



รูปที่ 3.10 วงจรไมโครคอนโทรลเลอร์

3.2.2 วงจรสวิตช์ควบคุมแรงดัน

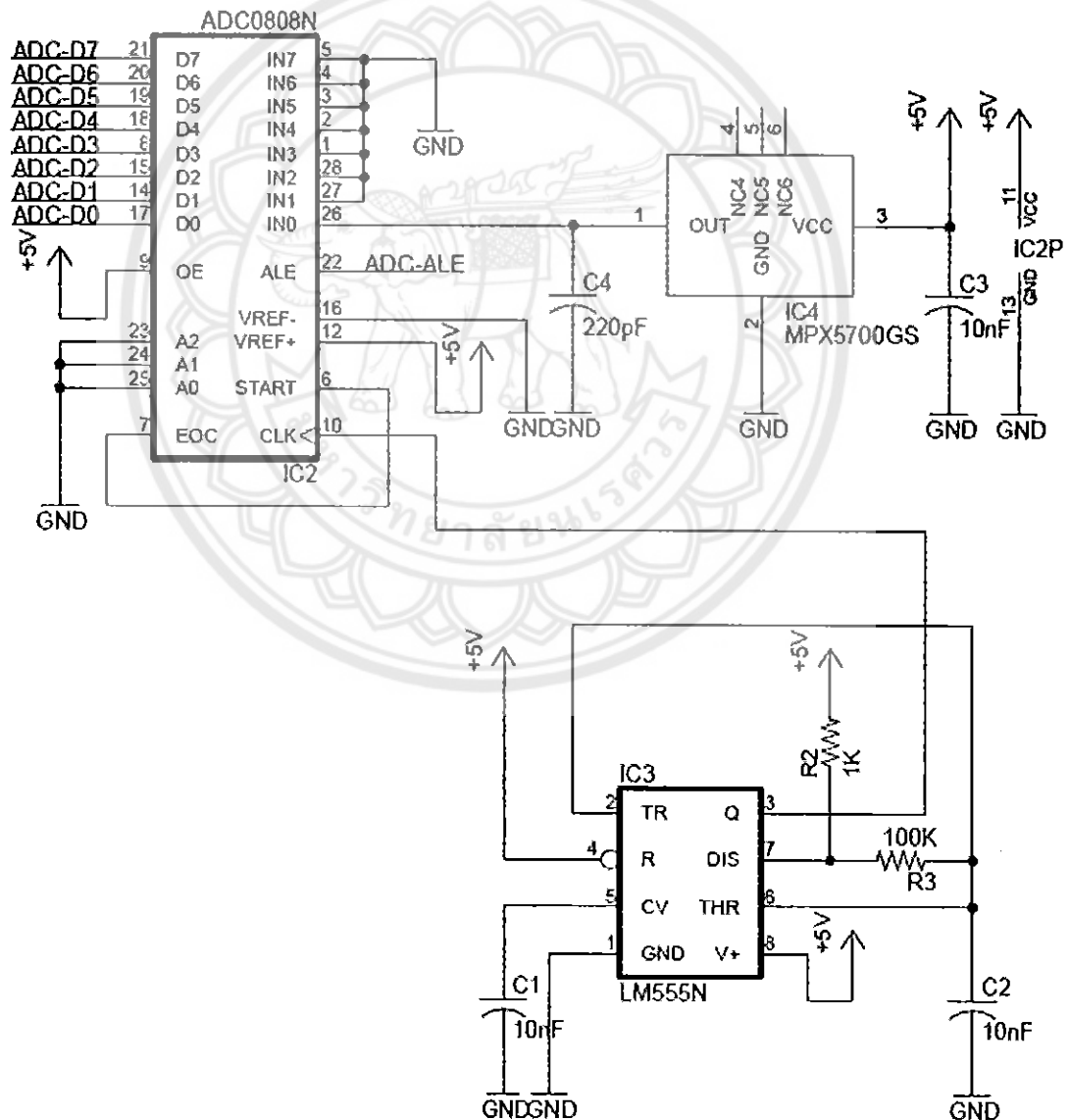
วงจรมีเป็นวงจรสวิตช์สำหรับ เพิ่มและลด แรงดันให้เป็นไปตามที่เราต้องการ โดยต่อกับพอร์ต C ที่ขา RC0 และ RC1 ของไมโครคอนโทรลเลอร์ โดยแสดงดังรูปที่ 3.11



รูปที่ 3.11 วงจรสวิตช์ควบคุมแรงดัน

3.2.3 วงจรเซนเซอร์ตรวจวัดแรงดัน

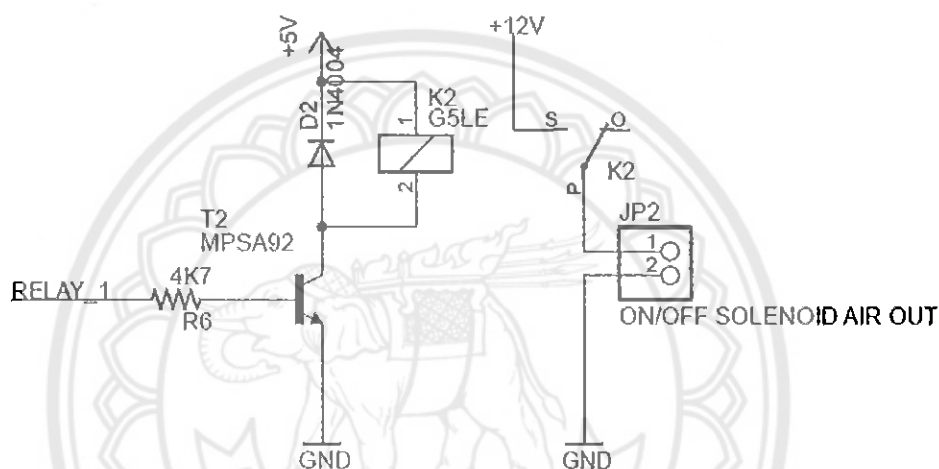
วงจรมีเซนเซอร์ตรวจวัดแรงดันที่เข้ามาจากสายเคเบิล แล้วเซนเซอร์จะทำการส่งค่าแรงดันที่เป็นสัญญาณอนาล็อกให้กับ ADC0808 8 บิต ทำหน้าที่แปลงอนาล็อกเป็นดิจิทัล โดยมีไอซี555 ต่อเข้ากับขา CLK ของ ADC0808 ทำงานในโหมดโมโนสเตเบิล ซึ่งเป็นการทำงานแบบซิงเกิ้ลช็อต หรือวันช็อต โดยการสร้างสัญญาณครั้งเดียวนี้ ทำหน้าที่เป็นไทม์เมอร์ช่วยตรวจสอบสัญญาณพัลส์ที่ส่งจากเซนเซอร์ หลังจากนั้น ADC0808 จะส่งสัญญาณค่าแรงดันที่แปลง ไปให้ไมโครคอนโทรลเลอร์ เพื่อทำการคำนวณค่า แล้วแสดงค่าทางหน้าจอแสดงผลแบบดิจิทัลต่อไป โดยแสดงดังรูปที่ 3.12



รูปที่ 3.12 วงจรเซนเซอร์ตรวจวัดแรงดัน

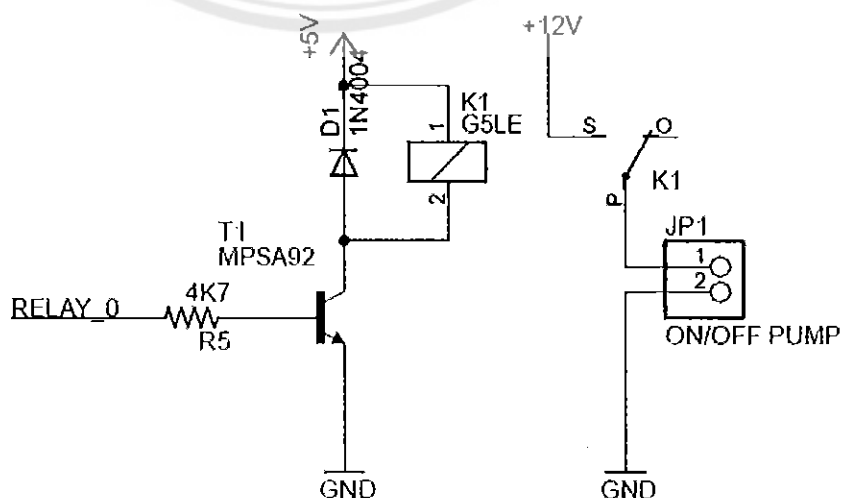
3.2.4 วงจรควบคุมโซลินอยด์วาล์ว

วงจรนี้ใช้รีเลย์ซึ่งทำหน้าที่ ตัด/ต่อไฟเพื่อเปิด/ปิดโซลินอยด์วาล์ว โดยต่อกับพอร์ต A ที่ขา RA1 ของไมโครคอนโทรลเลอร์ เมื่อเซนเซอร์วัดค่าได้มากกว่าที่เราต้องการ จะส่งค่าไปให้ไมโครคอนโทรลเลอร์และไมโครคอนโทรลเลอร์จะส่งให้รีเลย์ทำการต่อไฟให้กับโซลินอยด์วาล์ว โซลินอยด์วาล์วจะปล่อยลม เมื่อเซนเซอร์วัดได้ค่าตามที่เรากำหนดก็จะส่งค่าไปให้ไมโครคอนโทรลเลอร์และไมโครคอนโทรลเลอร์จะส่งให้รีเลย์ตัดไฟ โซลินอยด์วาล์วจะไม่ทำงาน โดยแสดงดังรูปที่ 3.13



รูปที่ 3.13 วงจรควบคุมโซลินอยด์วาล์ว

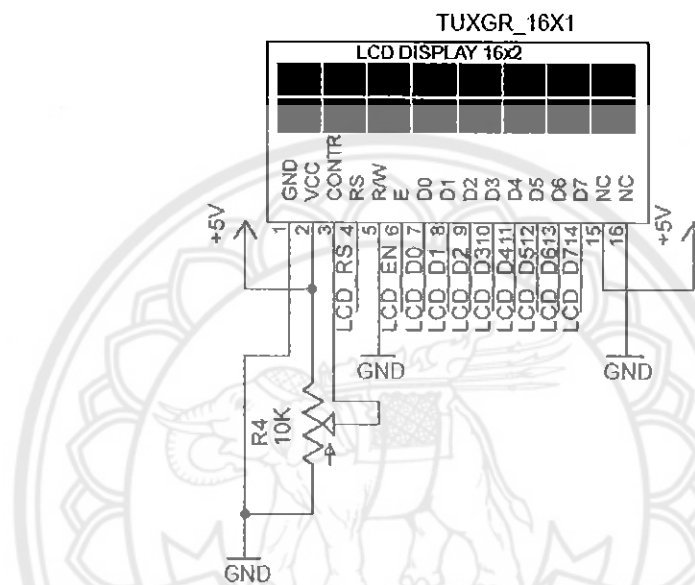
3.2.5 วงจรควบคุมเครื่องปั๊มลม



รูปที่ 3.14 วงจรควบคุมเครื่องปั๊มลม

จากรูปที่ 3.14 วงจรนี้ใช้รีเลย์ซึ่งทำหน้าที่ตัด/ต่อไฟเพื่อเปิด/ปิดวงจร โดยต่อกับพอร์ต A ที่ขา RA0 ของตัวไมโครคอนโทรลเลอร์ โดยในสภาวะปกติรีเลย์จะต่อไฟส่งสัญญาณให้เครื่องปั๊มทำงาน เมื่อเซนเซอร์วัดค่าได้ตามที่ต้องการจะส่งค่าไปให้ไมโครคอนโทรลเลอร์และไมโครคอนโทรลเลอร์จะสั่งให้รีเลย์ตัดไฟ เครื่องปั๊มก็จะไม่ทำงาน

3.2.6 วงจรภายในหน้าจอแสดงผลแบบดิจิทัล

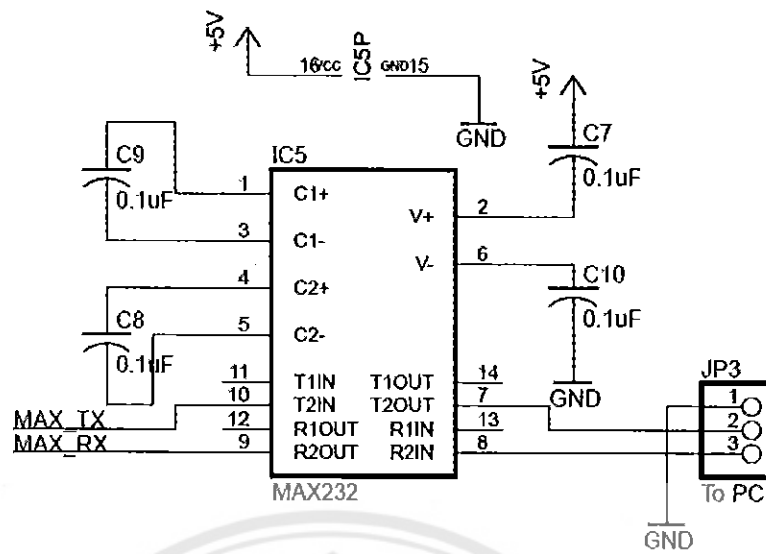


รูปที่ 3.15 วงจรหน้าจอแสดงผลแบบดิจิทัล

จากรูปที่ 3.15 ทำหน้าที่แสดงผลตัวเลข ตัวเลข และหน่วยโดยต่อกับพอร์ต C ที่ขา RC2, RC3 และพอร์ต D ซึ่งมีรูปแบบในการทำงาน S XXXA XXX PSI โดยจะรับค่าแรงดันจากไมโครคอนโทรลเลอร์มาแสดงผลเป็นดิจิทัล จะแสดงทั้งค่าที่วัดได้ และค่าที่เรากำหนด

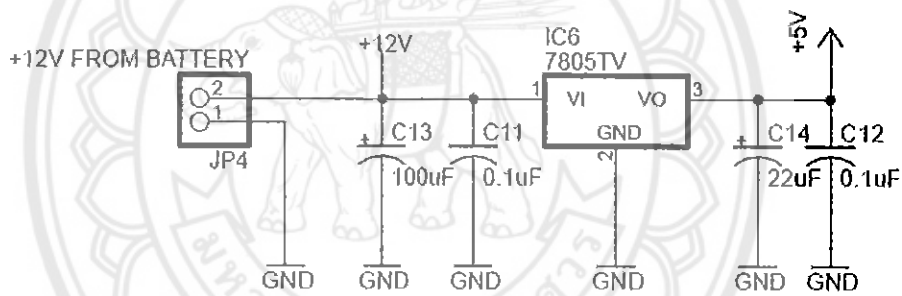
3.2.7 วงจรไอซีเบอร์ MAX232

MAX232 ทำตัวปรับระดับสัญญาณ จากระดับที่ทีแอล (TTL) ให้เป็นไปตามมาตรฐานของ RS-232 เพราะสัญญาณที่ออกมาจากพอร์ตของตัวไมโครคอนโทรลเลอร์นั้นเป็นสัญญาณระดับทีแอล (TTL) ซึ่งจะมีระดับสัญญาณอยู่ที่ 0-5 โวลต์ แต่การติดต่อสื่อสารในพอร์ตแบบอนุกรมตามมาตรฐาน RS-232 นั้นสัญญาณ Logic "0" ต้องมีระดับสัญญาณอยู่ที่ 3-15 โวลต์ และลอจิก "1" ต้องมีระดับสัญญาณอยู่ที่ (-3)-(-15) โวลต์ ดังนั้นในการใช้งานสื่อสารตามมาตรฐาน RS-232 เราจึงต้องใช้วงจรสื่อสารผ่านพอร์ตอนุกรมซึ่งมีไอซี MAX232 โดยแสดงดังรูปที่ 3.16



รูปที่ 3.16 วงจรไอซีเบอร์ MAX232

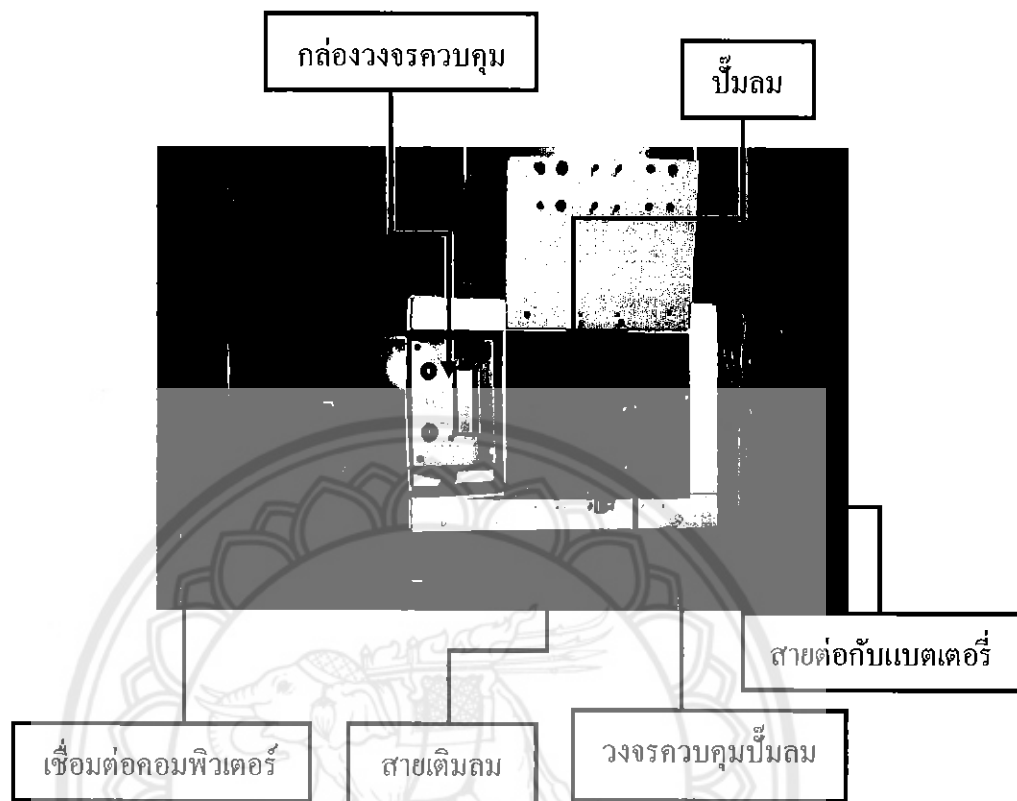
3.2.8 วงจรเบตเตอร์



รูปที่ 3.17 วงจรเบตเตอร์

จากรูปที่ 3.17 เบตเตอร์ +12 โวลต์ ทำหน้าที่จ่ายไฟให้กับไมโครคอนโทรลเลอร์ และจ่ายผ่านรีเลย์เพื่อจ่ายไฟให้กับปั๊มลม โซลินอยด์วาล์ว

3.3 โครงสร้างเครื่องเติมลมอัตโนมัติ



รูปที่ 3.18 รูปแสดงโครงสร้างเครื่องเติมลม

จากรูปที่ 3.18 เครื่องเติมลมจะประกอบด้วย 6 ส่วนด้วยกันคือ

1. กล่องวงจรวัดลม โดยภายในจะประกอบด้วยวงจรไมโครคอนโทรลเลอร์และวงจรวัดน้ำจอยแสดงผล
2. ปั๊มลม
3. สายต่อแบตเตอรี่
4. วงจรวัดลม โดยจะประกอบด้วย โซลินอยด์วาล์ว เซนเซอร์ และรีเลย์ ซึ่งต่อกับเครื่องปั๊มลมและไมโครคอนโทรลเลอร์
5. สายเติมลม
6. สายเชื่อมต่อกับคอมพิวเตอร์

บทที่ 4

ผลการทดลองและผลการวิเคราะห์

หลังจากสร้างเครื่องเติมลมอัตโนมัติ ผู้ดำเนินโครงการได้ดำเนินการทดสอบการทำงานของเครื่องเติมลมอัตโนมัติ โดยแบ่งการทดสอบออกเป็น 3 ส่วนการทดลองดังนี้

1. การทดลองความสามารถในการเติมลมผ่านการควบคุมด้วยมือ
2. การทดลองความสามารถในการเติมลมผ่านการควบคุมด้วยโปรแกรมวิซวลเบสิก
3. การทดลองระยะเวลาที่ใช้ในการเติมลม

4.1 การทดลองความสามารถในการเติมลมผ่านการควบคุมด้วยมือ

ในการทดลองความสามารถในการเติมลมผ่านการควบคุมด้วยมือ จะทดลองโดยที่กำหนดแรงดันเริ่มต้นที่ 10 ปอนด์ต่อตารางนิ้ว แล้วทำการเติมให้ได้แรงดัน 21 – 40 ปอนด์ต่อตารางนิ้ว โดยการต่ออุปกรณ์ ดังรูปที่ 4.1 และมีผลการทดลองแสดงดังตารางที่ 4.1



รูปที่ 4.1 รูปแสดงการเติมลมผ่านการควบคุมด้วยมือ

ตารางที่ 4.1 แสดงผลการทดลองความสามารถในการเติมลมผ่านการควบคุมด้วยมือ

แรงดัน (PSI)	แรงดันที่ได้ ครั้งที่ (PSI)					แรงดัน เฉลี่ย
	1	2	3	4	5	
21	20	20.5	20.5	20.5	20.5	20.4
22	21.5	21.5	21.5	22	21.5	21.6
23	22.5	23	22.5	22.5	22.5	22.6
24	24	23.5	23.5	23.5	23.5	23.6
25	24.5	24.5	24.5	25	25	24.7
26	25.5	25.5	25.5	25.5	25.5	25.5
27	26.5	26.5	27	26.5	27	26.7
28	27.5	27.5	27.5	27.5	27.5	27.5
29	28.5	29	28.5	28.5	29	28.7
30	29.5	29.5	30	29.5	29.5	29.6
31	30.5	30.5	30.5	30.5	30.5	30.5
32	32	31.5	31.5	31.5	31.5	31.6
33	32.5	33	32.5	32.5	33	32.7
34	33.5	33.5	34	33.5	34	33.7
35	34.5	35	35	35	34.5	34.8
36	36	36	35.5	36	35.5	35.8
37	36.5	37	36.5	36.5	36.5	36.6
38	38	38	37.5	38	38	37.9
39	38.5	39	39	39	38.5	38.8
40	40	39.5	39.5	39.5	39.5	39.6

จากตารางที่ 4.1 ผลการทดลองการเติมลมผ่านการควบคุมด้วยมือทำให้ทราบว่าเครื่องเติมลมอัตโนมัติสามารถเติมลมได้แรงดันตั้งแต่ 0 ปอนด์ต่อตารางนิ้วถึงแรงดันสูงสุดที่ 40 ปอนด์ต่อตารางนิ้ว

หมายเหตุ : มีแรงดันลดลงจากค่าที่กำหนด 0.5 ปอนด์ต่อตารางนิ้ว เนื่องจากการถอดสายเติมลมของผู้ใช้

4.2 การทดสอบความสามารถในการเติมลมผ่านการควบคุมด้วยโปรแกรมวิชาการเบสิก

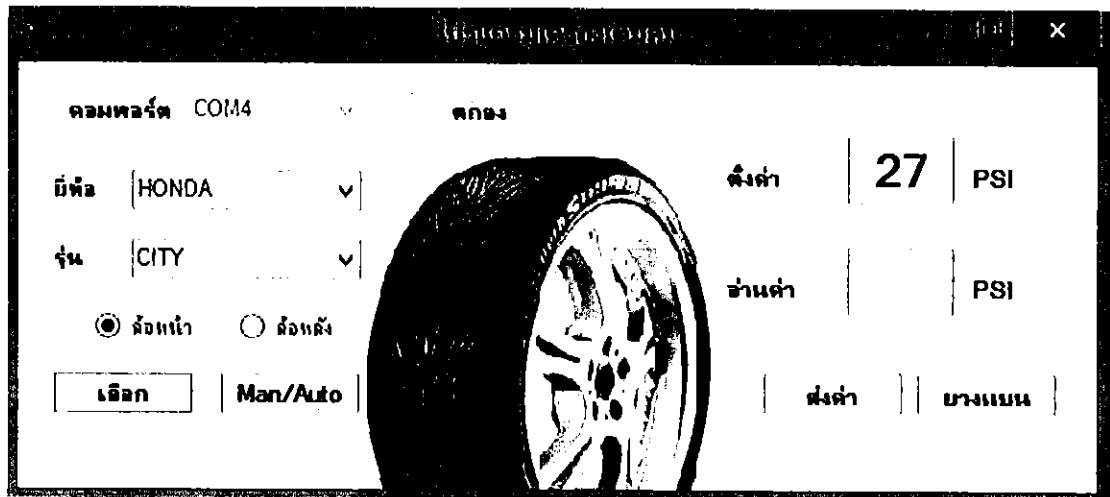
ในการทดสอบความสามารถในการเติมลมผ่านการควบคุมด้วยโปรแกรมวิชาการเบสิกจะทำการทดลองโดยแบ่งออกเป็นแบบอัตโนมัติและแบบป้อนค่า โดยการต่ออุปกรณ์ ดังรูปที่ 4.2



รูปที่ 4.2 รูปแสดงการเติมลมผ่านการควบคุมด้วยโปรแกรมวิชาการเบสิก

4.2.1 แบบอัตโนมัติ

จะทำการทดลองตามระดับแรงดันของรถแต่ละรุ่น ทำการทดลองเลือกการเติมลมล้อหน้า โดยมีแรงดันล้อหน้าของแต่ละรุ่น ที่กำหนดดังตารางที่ 4.2 และมีลักษณะการทำงานของโปรแกรม ดังรูปที่ 4.3



รูปที่ 4.3 รูปแสดงลักษณะการทำงานของ โปรแกรมแบบอัตโนมัติ

ตารางที่ 4.2 แสดงผลการทดสอบความสามารถในการเติมลมผ่านการควบคุมด้วยโปรแกรมวิซวล เบสิคแบบอัตโนมัติ

รุ่นรถ	แรงดันที่กำหนด	แรงดันที่ได้ ครั้งที่ (PSI)					แรงดันเฉลี่ย
		1	2	3	4	5	
Honda City	27	26.5	26.5	26.5	26.5	27	26.6
Honda Civic	30	30	29.5	29.5	29.5	30	29.7
Honda Accord	33	33	33	33	32.5	32.5	32.8
Toyota Vios	32	31.5	32	31.5	31.5	32	31.7
Toyota Vigo	32	31.5	31.5	31.5	31.5	31.5	31.5
Toyota Camry	32	32	32	31.5	31.5	31.5	31.7
Nissan Sunny	32	31.5	31.5	31.5	32	32	31.7
Nissan Teana	32	32	32	32	31.5	31.5	31.8
Nissan March	32	31.5	32	31.5	32	31.5	31.7

ตารางที่ 4.2 (ต่อ) แสดงผลการทดลองความสามารถในการเติมลมผ่านการควบคุมด้วยโปรแกรม
วิซวลเบสิกแบบอัตโนมัติ

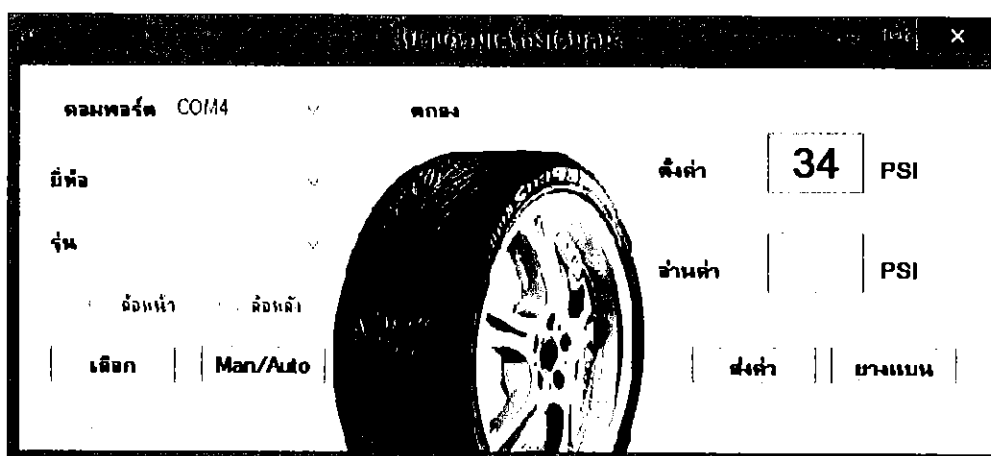
รุ่นรถ	แรงดันที่กำหนด	แรงดันที่ได้ ครั้งที่ (PSI)					แรงดันเฉลี่ย
		1	2	3	4	5	
Mitsubishi Lancer	29	29	28.5	28.5	28.5	28.5	28.6
Mitsubishi Attranger	30	29.5	30	30	29.5	30.5	29.7
Mitsubishi Pajero	32	31.5	31.5	31.5	31.5	31.5	31.5

จากตารางที่ 4.2 ผลการทดลองการเติมลมผ่านการควบคุมด้วยโปรแกรมวิซวลเบสิกแบบอัตโนมัติ ทำให้ทราบว่าเครื่องเติมลมอัตโนมัติสามารถเติมลมตามแรงดันของแต่ละรุ่นที่กำหนดผ่านโปรแกรมวิซวลเบสิกได้

หมายเหตุ : มีแรงดันลดลงจากค่าที่กำหนด 0.5 ปอนด์ต่อตารางนิ้ว เนื่องจากการถอดสายเติมลมของผู้ใช้

4.2.2 แบบป้อนค่า

จะทำการทดลองโดยที่กำหนดแรงดันเริ่มต้นที่ 10 ปอนด์ต่อตารางนิ้วแล้วทำการเติมให้ได้แรงดัน 21 – 40 ปอนด์ต่อตารางนิ้ว ผลการทดลองแสดงได้ดังตารางที่ 4.3 และมีลักษณะการทำงานของโปรแกรมดังรูปที่ 4.4



รูปที่ 4.4 รูปแสดงลักษณะการทำงานของโปรแกรมแบบป้อนค่า

ตารางที่ 4.3 แสดงผลการทดลองความสามารถในการเติมลมผ่านการควบคุมด้วย โปรแกรม
วิซวลเบสิกแบบป้อนค่า

แรงดัน (PSI)	แรงดันที่ได้ ครั้งที่ (PSI)					แรงดัน เฉลี่ย
	1	2	3	4	5	
21	20.5	20.5	20.5	20.5	20.5	20.5
22	21.5	21.5	21.5	21.5	22	21.6
23	23	22.5	22.5	22.5	22.5	22.6
24	23.5	24	24	23.5	23.5	23.7
25	24.5	25	24.5	24.5	24.5	24.6
26	25.5	25.5	25.5	26	26	25.7
27	26.5	26.5	26.5	26.5	26.5	26.5
28	27.5	28	27.5	28	28	27.8
29	29	29	28.5	28.5	28.5	28.7
30	29.5	29.5	30	29.5	29.5	29.6
31	31	30.5	30.5	30.5	30.5	30.6
32	32	31.5	31.5	31.5	32	31.7
33	32.5	32.5	33	32.5	32.5	32.6
34	33.5	33.5	34	34	34	33.8
35	34.5	34.5	34.5	34.5	34.5	34.5
36	35.5	36	35.5	35.5	36	35.7
37	37	37	36.5	36.5	36.5	36.7
38	37.5	37.5	37.5	38	38	37.7
39	39	39	39	38.5	39	38.9
40	40	39.5	39.5	40	40	39.8

จากตารางที่ 4.3 ผลการทดลองการเติมลมผ่านการควบคุมด้วย โปรแกรมวิซวลเบสิกแบบ
ป้อนค่าทำให้ทราบว่าเครื่องเติมลมอัตโนมัติสามารถเติมลมได้แรงดันตั้งแต่ 0 ปอนด์ต่อตารางนิ้วถึง
แรงดันสูงสุดที่ 40 ปอนด์ต่อตารางนิ้ว

หมายเหตุ : มีแรงดันลดลงจากค่าที่กำหนด 0.5 ปอนด์ต่อตารางนิ้ว เนื่องจากการถอดสายเติมลมของ
ผู้ใช้

4.3 การทดลองระยะเวลาที่ใช้ในการเติมลม

จะทดลองโดยที่กำหนดแรงดันเริ่มต้นที่ 10 ปอนด์ต่อตารางนิ้วแล้วทำการเติมให้ได้แรงดัน 21-40 ปอนด์ต่อตารางนิ้วและทำการจับเวลาในการเติมลมแต่ละครั้ง ผลการทดลองแสดงดังตารางที่ 4.4

ตารางที่ 4.4 แสดงผลการทดลองระยะเวลาที่ใช้ในการเติมลม

แรงดัน (PSI)	เวลาที่ใช้ ครั้งที่ (นาที)					เวลา เฉลี่ย
	1	2	3	4	5	
21	1.35	1.33	1.40	1.35	1.35	1.36
22	1.48	1.40	1.45	1.48	1.48	1.46
23	1.50	1.52	1.50	1.51	1.49	1.50
24	1.53	1.54	1.53	1.52	1.54	1.53
25	2.11	2.10	2.11	2.12	2.11	2.11
26	2.36	2.35	2.36	2.36	2.34	2.35
27	2.37	2.38	2.37	2.35	2.37	2.37
28	2.58	2.56	2.58	2.57	2.56	2.57
29	2.55	2.54	2.54	2.56	2.57	2.55
30	3.05	3.10	3.05	3.09	3.10	3.07
31	3.30	3.35	3.32	3.37	3.33	3.33
32	3.32	3.30	3.32	3.34	3.32	3.32
33	3.39	3.39	3.37	3.38	3.37	3.38
34	3.59	3.59	3.58	3.58	3.59	3.59
35	4.04	4.02	4.04	4.05	4.02	4.03
36	4.00	4.05	4.07	4.05	4.05	4.04
37	4.08	4.08	4.06	4.08	4.06	4.07
38	4.14	4.12	4.14	4.10	4.12	4.12
39	5.00	5.02	5.01	5.04	5.00	5.01
40	5.15	5.15	5.17	5.16	5.15	5.16

จากตารางที่ 4.4 ผลการทดลองระยะเวลาที่ใช้ในการเติมลมจะเห็นว่าที่แรงดัน 21-24 ปอนด์ต่อตารางนิ้วจะใช้เวลาประมาณ 1.30-2 นาที ที่แรงดัน 25-29 ปอนด์ต่อตารางนิ้วจะใช้เวลาประมาณ 2-3 นาที ที่แรงดัน 31-34 ปอนด์ต่อตารางนิ้วจะใช้เวลาประมาณ 3.30-4 นาที ที่แรงดัน 35-38 ปอนด์ต่อตารางนิ้วจะใช้เวลาประมาณ 4 นาที และที่แรงดันสูงสุด 40 ปอนด์ต่อตารางนิ้ว จะใช้เวลาในการเติมประมาณ 5 นาที โดยระยะเวลาที่ใช้ในการเติมมากหรือน้อยขึ้นอยู่กับแรงดัน ที่แรงดันสูงจะใช้เวลาในการเติมนานขึ้น



บทที่ 5

สรุปผลและข้อเสนอแนะ

จากการศึกษา ออกแบบ ประกอบ ทดสอบ และทำการปรับปรุงชิ้นงานจนเป็นเครื่องเติมลมอัตโนมัติโดยใช้ระยะเวลาดำเนินโครงการ 2 ภาคการศึกษา ทำให้ทราบถึงปัญหาที่เกิดขึ้นในการทำโครงการ สำหรับบทนี้จะเป็นการสรุปผลที่ได้จากการทดลองในโครงการ พร้อมเสนอแนะแนวทางในการนำโครงการนี้ไปพัฒนาหรือต่อยอดให้มีประสิทธิภาพมากยิ่งขึ้น

5.1 สรุปผลการทดลองการทำงานของเครื่องเติมลมอัตโนมัติ

ในการทำโครงการเครื่องเติมลมอัตโนมัติ เป็นการสร้างโครงการเพื่อการศึกษาและสามารถนำไปใช้ประโยชน์ได้จริง สามารถสรุปผลการทดลองการทำงานของเครื่องเติมลมอัตโนมัติได้ดังนี้

1. การทดลองการเติมลมผ่านการควบคุมด้วยมือ เครื่องเติมลมสามารถเติมลมแรงดันสูงสุดถึง 40 ปอนด์ต่อตารางนิ้วได้
2. การทดลองการเติมลมผ่านการควบคุมด้วยโปรแกรมวิซวลเบสิก เครื่องเติมลมสามารถเติมลมได้จริง ตามการควบคุมผ่าน โปรแกรมวิซวลเบสิกทั้งแบบอัตโนมัติและแบบป้อนค่า
3. การทดลองระยะเวลาที่ใช้ในการเติมลม เครื่องเติมลมอัตโนมัติสามารถเติมลมจากแรงดันเริ่มต้นที่ 10-40 ปอนด์ต่อตารางนิ้วได้ โดยใช้เวลาในการเติมประมาณ 5 นาที และระยะเวลาที่ใช้ในการเติมจะขึ้นอยู่กับแรงดัน

คือ ถ้าแรงดันยิ่งสูงก็จะยิ่งใช้เวลาในการเติมมาก

5.2 ปัญหาและแนวทางแก้ไขปัญหา

1. ปัญหา : ในการเติมลมหากผู้ใช้ทำการถอดสายเติมซ้ำเกินไป จะทำให้เกิดลมรั่วออกมาก ซึ่งส่งผลให้ค่าของแรงดันมีความคลาดเคลื่อนสูง

การแก้ไข : เปลี่ยนหัวที่ใช้ในการเติมลมที่ทำให้สามารถถอดง่าย จะช่วยให้แรงดันที่เติมมีความคลาดเคลื่อนน้อยลง

2. ปัญหา : ในการเติมลมที่แรงดันสูงๆ ใช้เวลานานเกินไป

การแก้ไข : หาปั๊มลมที่สามารถปั๊มลมได้เร็วและแรงขึ้น

3. ปัญหา : ปั๊มที่ใช้ในการกดเพิ่ม/ลดแรงดันในการเติมลมแบบควบคุมด้วยมือกดยากทำให้การเพิ่ม/ลดแรงดันทำได้ลำบาก

การแก้ไข : หาสวิทช์ปั๊มกดที่ง่ายต่อการกด และมีการตอบสนองที่เร็วกว่าเดิม

4. ปัญหา : หน้าจอแสดงผลยังมีความไม่เสถียร เพราะมีสัญญาณรบกวนจากการทำงานของโปรแกรม

การแก้ไข : ต่อตัวเก็บประจุบายพาสที่แหล่งจ่ายไฟ เพื่อช่วยในการลดอิมพีแดนซ์ของสาย และช่วยลดสัญญาณรบกวนจากภายในและภายนอก

5. ปัญหา : โครงสร้างภายนอกเป็นกระดาษ หากโดนความชื้นจะทำให้โครงสร้างเสียหายและส่งผลกระทบต่ออุปกรณ์ภายใน

การแก้ไข : เปลี่ยนวัสดุที่ใช้ในการทำโครงสร้างให้มีความทนทานมากขึ้น อาจใช้เป็นอะคริลิกหรือพลาสติก

5.3 แนวทางการพัฒนาโครงการงาน

เพื่อให้สามารถนำโครงการนี้ไปพัฒนา หรือต่อยอดให้มีประสิทธิภาพในการใช้งานได้ดีขึ้น จึงมีแนวทางการพัฒนาโครงการ ดังนี้

1. พัฒนาหัวที่ใช้ในการเติมลมให้สามารถถอดเปลี่ยนได้ และมีหัวเติมหลายแบบ ให้สะดวกต่อการใช้งาน และมีการทำงานหลากหลายยิ่งขึ้น

2. เลือกใช้สวิทช์กดที่มีการตอบสนองไว เพื่อให้สะดวกต่อการใช้งาน

3. เพิ่มโปรแกรมสำหรับบันทึกค่า ในการเติมลมแต่ละครั้ง เพื่อช่วยให้ผู้ใช้วางแผนในการดูแลยางรถยนต์ได้

เอกสารอ้างอิง

- [1] ทฤษฎีลม. สืบค้นเมื่อ 6 สิงหาคม 2556
จาก https://www.google.co.th/cic.wu.ac.th/Data_Download/Research/ลม.pdf
- [2] การตรวจวัดความดัน. สืบค้นเมื่อ 6 สิงหาคม 2556
จาก <http://somsak.me.engr.tu.ac.th/download/flow%20rate.pdf>
- [3] นิวเมติกส์. สืบค้นเมื่อ 6 สิงหาคม 2556
จาก https://utcc2.utcc.ac.th/engineer/learning/chalermchon_vis/.../Automation01.pdf
- [4] เซนเซอร์. สืบค้นเมื่อ 6 สิงหาคม 2556
จาก http://www.cpe.ku.ac.th/~yuen/204471/sensor/temp_pres
- [5] ไมโครคอนโทรลเลอร์PIC 16F877. สืบค้นเมื่อ 6 สิงหาคม 2556
จาก <http://elec-thai.blogspot.com/2012/12/pic.html>
- [6] โปรแกรมไมโครซอฟท์วิซวลเบสิก (Microsoft Visual Basic). สืบค้นเมื่อ 6 สิงหาคม 2556
จาก <http://th.wikipedia.org/wiki/วิซวลเบสิก>
- [7] โซลินอยด์วาล์ว. สืบค้นเมื่อ 11 สิงหาคม 2556
จาก <http://webserv.kmitl.ac.th/s1010958/web/php/Solenoidvalves.php>
- [8] การเติมลมยางที่เหมาะสมของรถยนต์. สืบค้นเมื่อ 11 สิงหาคม 2556
จาก <http://www.sakao.go.th/CAR1.htm>



```

#include <htc.h>
#include <stdio.h>
#include "delay.h"
#include "initiallcd.h" __CONFIG(HS & WDTDIS & PWRTDIS & BORDIS & LVPDIS &
WRTEN & DEBUGDIS & DUNPROT & UNPROTECT);
#define DOWN_PRESS RC0 == 0 //
#define UP_PRESS RC1 == 0 //
#define ALE_HIGH RC5 = 1 //
#define ALE_LOW RC5 = 0 //
#define init_80adc() ALE_HIGH; delay_ms(10); ALE_LOW //
#define BEEB_ON RA2 = 1 // กำหนดตัวแปรต่างๆ
#define BEEB_OFF RA2 = 0 //
//
#define AIR_IN_ON RA0 = 1 //
#define AIR_IN_OFF RA0 = 0 //
//
#define AIR_OUT_ON RA1 = 1 //
#define AIR_OUT_OFF RA1 = 0 //
void interrupt my_isr(void); //
void usart_string(unsigned char*s); // วอยด์ที่จะนำไปใช้
void sw_check(void); //
char buffer[9]; //

char buffer1[9]; // ประกาศตัวแปร(ความจุ 1 ไบต์)บัพเฟอร์เพื่อ
// นำไปใช้ในการแสดงผลบนหน้าจอแอลซีดี
char buffer2[9]; //
char buff_data[5]; //
unsigned char index = 0; //
unsigned long set_value = 35; // เซทค่าเริ่มต้นที่ 35
unsigned long actual_value = 0; // ค่าจริงเริ่มต้นที่ 0
unsigned long adc_value = 0; // เอดีซีค่าที่อ่านได้เริ่มต้นที่ 0
unsigned long adc_value_new = 0; // เอดีซีค่าใหม่ที่ปรับตั้งเริ่มต้นที่ 0

```

```

unsigned long adc_psi = 0; // เอ็ดซีพีเอสไอเริ่มต้นที่ 0
unsigned char adc_uart = 0; // เอ็ดซียูอาร์ทีเริ่มต้นที่ 0
double MM = 0.0463149416; //
double CC = 0.0990334273; // กำหนดตัวแปรและค่าของแต่ละตัวแปร ใน
// การคำนวณความดันของเซนเซอร์เป็นแรงดัน
// ส่งเข้าคอนโทรลเลอร์ และแสดงผล

double ST = 0.0196078431; //
bitflag_init = 0; // บิตอนินิเซียลใช้เชื่อมกับคอมพิวเตอร์เริ่มต้น
ที่ 0
bitflag_noair = 0; // บิตโนแอร์ใช้สำหรับในกรณียางแบนหรือไม่
// มีความดันอยู่เลยภายในยาง

double volt; // กำหนดตัวแปรใช้คำนวณ โวลต์

void main(void)
{
    delay_ms(500); initial_lcd(); //
    initial LCD // เซทให้เอ็ดซีดีเตรียมรับค่า
    TRISA = 0x00; //
    PORTA is Output // ประกาศไดเรกพอร์ต พอร์ตเอเป็นเอาต์พุต
    TRISB = 0xFF; //
    PORTB is input // ประกาศไดเรกพอร์ต พอร์ตบีเป็นอินพุต
    TRISC5 = 0; //
    PORTC5 is output // ประกาศไดเรกพอร์ต พอร์ตซีห้าเป็นเอาต์พุต
    RC5 = 1; //
    PORTC5 is High // พอร์ตซีห้าเป็นไฮท์
    init_80adc(); //
    initial ADC0808 // อินนิเซียล เอ็ดซี0808 แปลงอนาล็อกเป็นดิจิตอล
    TXSTA = 0x24; //
    TX Enable 8 Asynchronous High speed //
    initialuart and interrupt // เปิดการใช้งานที่เอ็ดกซ์8 บิตอะซิงโครนัสไฮท์สปีด
    // เซทค่ายูอาร์ทีและอินเตอร์รัพท์
    RCSTA = 0x90; //

```

```

Serial Port Enable and continuous // เปิดการใช้งานพอร์ตอนุกรมและต่อเนื่องด้วยคำสั่งนี้
SPBRG = 103; //
baud rate 9600 at 16MHz // เซทบอร์ดเรท9600 ที่ความถี่ 16 เมกกะเฮิร์ต จำนวนค่าได้ 103
PEIE = 1; //
Peripheral Interrupt Enable bit // อุปกรณ์ต่อพ่วงขัดจังหวะเพื่อเปิดใช้งานบิต
RCIE = 1; //
RX interrupt enable // โปรแกรมคอมพิวเตอร์ทำงานส่งค่ายูอาร์ทีมาคอนโทรลเลอร์
// จะหยุดทำงานในเมนและกระโดดไปทำงานในอินเตอร์รัพท์แทน
RCIF = 0; //
flag RX interrupt // และพอทำงานเสร็จก็จะ โดคกลับเข้ามา
ei0; //
enableglobe interrupt // ช่วยขัดจังหวะทั้งหมด
while(1)
{
addr_lcd_xy(1,1); // แสดงผลบนหน้าจอแอลซีดี
"S XXX A" // ค่าที่เซท
sprintf(buffer,"S %3u A",set_value); //
string_lcd(buffer); //
init_80adc0; // เซทให้เอดีซี0808 เตรียมรับค่า
adc_value = PORTB; // อ่าน 8 บิตจากเอดีซี0808 for(int j = 0; j < 8000 ; j++)
//
{
//
delay_us(30); //
adc_value_new = PORTB; //
if(adc_value>adc_value_new) // อ่าน 8 บิตจากเอดีซี0808 และเลือกค่าต่ำสุดของค่า
8000
{
//
adc_value = adc_value_new; //
}
//
}
//
volt = adc_value * ST; // V = (8 bit ADC) * (5/255) การคำนวณ โดยอ้างอิงจาก
// ค่าต่ำซีทของเซนเซอร์

```



```

actual_value = (volt - CC)/MM; // Y = mX + C // X = (Y-C)/m
// PSI = (V - 0.0990334273) / 0.0463149416

if(actual_value <= 1) //
{ //
AIR_IN_OFF; // ถ้าค่าจริงน้อยกว่าหรือเท่ากับ 1 แล้วค่าจริงจะเป็น 0
actual_value = 0; //
} //
addr_lcd_xy(1,2); //
sprintf(buffer1, "%3u PSI", actual_value); // แสดงผลบนหน้าจอแอลซีดี "XXX PSI" // ค่า
จริง string_lcd(buffer1); //
if(actual_value != adc_uart) //
{ //
adc_uart = actual_value; // ถ้าค่าจริงเปลี่ยนแปลงแล้วส่งค่าไปยังคอมพิวเตอร์
sprintf(buffer2, "%3u", adc_uart); //
usart_string(buffer2); //
}
if(flag_init == 1) // สถานะของยูอาร์ท
{ //
flag_init = 0; // คอมพิวเตอร์เปิดพอร์ตและส่งค่าจริงกลับมาที่
คอนโทรลเลอร์
usart_string(buffer2); //
} //

if((actual_value < set_value) && (actual_value > 1)) //
{ // ค่าที่เซทมากกว่าค่าจริงที่ได้
AIR_IN_ON; //
}

if(actual_value > set_value) //
{ //
AIR_IN_OFF; //
delay_ms(100); //
AIR_OUT_ON; // ค่าที่เซทน้อยกว่าค่าจริงที่ได้
delay_ms(1500); //
}

```

```

    AIR_OUT_OFF; //
    delay_ms(100); //
}
if(actual_value == set_value) //
{ //
    AIR_IN_OFF; //
    AIR_OUT_OFF; // ค่าที่เซตเท่ากับค่าจริงที่ได้
    BEEB_ON; //
    delay_ms(500); //
    BEEB_OFF; //
}
if(flag_noair == 1) //
{ //
    delay_ms(100); //
    AIR_IN_ON; //
    delay_ms(7000); // คลิกปุ่มยางแบนจากโปรแกรมคอมพิวเตอร์
    // แล้วเครื่องจะเติมเองทั้งหมด 7 วินาที
    AIR_IN_OFF; //
    delay_ms(100); //
    //
    flag_noair = 0; //
} //
sw_check(); // สวิตช์เช็คฟังก์ชัน
}
}

voidsw_check(void)
{
    while(UP_PRESS) //
    { //
        delay_ms(10); //
        while(UP_PRESS) //

```

```

        { // อ่านสวิตช์ที่กดและเพิ่มค่าจากค่าที่ตั้งไว้
if(set_value<99) //
    { //
    set_value++; //
    } //
    addr_lcd_xy(1,1); //
    sprintf(buffer,"S %3d A",set_value); //
    string_lcd(buffer); //
    //
    delay_ms(150); //
    } //
    delay_ms(10); //
    } //
while(DOWN_PRESS) //
    { //
    delay_ms(10); //
    //
while(DOWN_PRESS) //
    { //
    //
if(set_value> 0) // อ่านสวิตช์ที่กดและลดค่าจากค่าที่ตั้งไว้
    { //
    set_value--; //
    } //
    //
    addr_lcd_xy(1,1); //
    sprintf(buffer,"S %3d A",set_value); //
    string_lcd(buffer); //
    //
    delay_ms(150); //
    } //
    delay_ms(10); //

```

```

    }
}

void interrupt my_isr(void) // ฟังก์ชันอินเทอร์รัพท์
{
    if(RCIE && RCIF) // เมื่ออาร์เอ็กซ์อินเทอร์รัพท์ทำงาน
    {
        buff_data[index] = RCREG; // อ่านตัวแปรไปยังอินเด็กซ์ อาร์เรย์
        if(buff_data[index] == 'x') // คอมพิวเตอร์เปิดพอร์ตส่งค่า'เอ็กซ์'
        {
            flag_init = 1; index = -1;
        }
        if(buff_data[index] == 'n') // คอมพิวเตอร์เปิดพอร์ตส่งค่า'เอน'
        {
            flag_noair = 1; index = -1;
        }
        if(buff_data[index] == '#') // หา '#' ที่สตริง
        {
            set_value = 0; switch (index)
            {
                case(1): set_value |= buff_data[index-1]-48; // ที่อยู่อาร์เรย์ของ '#' คือ 1
                break;
                case(2): set_value |= ((buff_data[index-2]-48)*10) + (buff_data[index-1]-48); // ที่อยู่
                break; // อาร์เรย์ของ '#' คือ 2

                default: set_value = 0; break;
            }
            index = -1;
        }
        index++; RCIF = 0; // เลิกรหัสอาร์เอ็กซ์อินเทอร์รัพท์
    }
}

void usart_string(unsigned char*s) // ยูอาร์ทีส่งสตริงฟังก์ชัน

```

```
}  
int i = 0; while (s[i] != 0)  
{  
while (TRMT == 0); TXREG = (s[i]); i++;  
}  
}
```





สภานวค ข

โครงการทำงานของโปรแกรมวิชาพลเบตค

มหาวิทยาลัยราชภัฏสกลนคร

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;           // การใช้ระบบอินพุตเอาท์พุตพอร์ต
namespace pump
{
public partial class Form1 : Form
{
public Form1()
{
InitializeComponent();
}
private void Form1_Load(object sender, EventArgs e) // ในส่วนของ
// หน้าต่างฟอร์มหนึ่ง
{
Load_ComPortList();
cbCar.Enabled = true;           // คอมโบบล็อคของรุ่นเป็นอินาเบิ้ล
cbComp.Enabled = true;         // คอมโบบล็อคของยี่ห้อเป็นอินาเบิ้ล
tB2.ReadOnly = true;           // แท็บล๊อกสองทำหน้าที่อ่านค่าอย่างเดียวใน
// ระบบอัตโนมัติ
}
private void List_load(object sender, EventArgs e) // การส่งค่าระหว่างคอม
// โบบล็อคจากยี่ห้อไปยังรุ่นของรถ
{
cbCar.Items.Clear();           // ให้คอมโบบล็อคสองเป็นไอเท็ม
stringselectedItem = cbComp.Items[cbComp.SelectedIndex].ToString();
}
}
}

```

```

if (selectedItem == "HONDA") //
{ //
cbCar.Items.Add("CITY"); // หากคอม โบบล็อกหนึ่งเลือกยี่ห้อ
// หากคอม โบบล็อกสองจะแสดงผล
// รุ่น ซิตี้ ชีวิค และแอกคอร์ด

cbCar.Items.Add("CIVIC"); //
cbCar.Items.Add("ACCORD"); //
}

else if (selectedItem == "TOYOTA") //
{ //
cbCar.Items.Add("VIOS"); // หากคอม โบบล็อกหนึ่งเลือกยี่ห้อ โต
// โยต้าคอม โบบล็อกสองจะแสดงผลรุ่น
// วีออส วิโก้และเกมรี่

cbCar.Items.Add("VIGO"); //
cbCar.Items.Add("CAMRY"); //
}

else if (selectedItem == "NISSAN") //
{ //
cbCar.Items.Add("SUNNY"); // หากคอม โบบล็อกหนึ่งเลือกยี่ห้อ นิส
// สันคอม โบบล็อกสองจะแสดงผลรุ่น
// ซันนี่ เทน่า และมาร์ท

cbCar.Items.Add("TEANA"); //
cbCar.Items.Add("MARCH"); //
}

else if (selectedItem == "MITSUBISHI") //
{ //
cbCar.Items.Add("LANCER"); // หากคอม โบบล็อกหนึ่งเลือกยี่ห้อ มิต
// ซูบิชิคอม โบบล็อกสองจะแสดงผลรุ่น
// เลนเซอร์ แอทเรนเจอร์ และปาเจโร

```



```

cbCar.Items.Add("ATTRANGER");           //
cbCar.Items.Add("PAJERO");              //
}
}

staticSerialPortcomPort;
stringinData = "";
Int32 intData = 0;
Int32 value_car_auto = 0;
int mode = 1;                           // โหมดศูนย์ให้เป็นแมนนวล , โหมด
                                         // หนึ่งให้เป็นออโต้
private void Load_ComPortList()         // เปิดพอร์ตคอนุกรมทั้งหมด
{
    cbSerial.Items.Clear();
    foreach (string name in SerialPort.GetPortNames())
    {
        cbSerial.Items.Add(name);       // ให้คอม โบบล็อกแสดงพอร์ต
    }
    try                                  //
    {                                     //
        cbSerial.SelectedIndex = 0;     // ตั้งคอม โบบล็อกให้สามารถเลือก
                                         // พอร์ตได้ตามที่ต้องการ
        btSelectSerial.Enabled = true;  // ตั้งปุ่มบันทึกให้สามารถกดตกลง
                                         // เชื่อมพอร์ตได้
    }
    catch                                //
    {                                     //
        MessageBox.Show("ค้นหาคอมพอร์ตไม่เจอ"); // ตรวจสอบพอร์ต หากไม่มีพอร์ต
                                         // เชื่อมต่ออยู่ จะมีบล็อกข้อความแจ้ง
                                         // ขึ้นมาเตือนว่า ค้นหาคอมพอร์ตไม่เจอ
        btSelectSerial.Enabled = false; //
    }                                     //
}

```

```

}

private void btSelectSerial_Click(object sender, EventArgs e)
// ในส่วนของปุ่มบันทึกทอน
// เมื่อคลิกปุ่มบันทึกทอนและส่งเอ็ชไปที่คอนโทรลเลอร์จากนั้นคอนโทรลเลอร์ก็
// จะส่งค่าจริงกลับมาแสดงที่ช่องแท็บลือคหนึ่ง
{
Try
{
comPort = new SerialPort(cbSerial.SelectedItem + "", 9600, Parity.None, 8,
StopBits.One); // อัตราการส่งข้อมูลของซีเรียลพอร์ตเท่ากับ
9600comPort.DataReceived += new
SerialDataReceivedEventHandler(comPort_DataReceived);
comPort.Open();
cbSerial.Enabled = false;
btSelectSerial.Enabled = false;
if (comPort.IsOpen) //
{ // คอมพอร์ตทำหน้าที่เป็นตัวแปรเอ็ชเชื่อมระหว่าง
// หน้าต่างโปรแกรมแสดงผลกับคอนโทรลเลอร์
comPort.Write("x"); //
} //
}
catch //
{ // ตรวจสอบพอร์ตถ้าค้นหาไม่เจอจะมีหน้าต่างข้อความ
// เตือนMessageBox.Show("ค้นหาคอมพอร์ตไม่เจอ"); //
} //
}

void comPort_DataReceived(object sender, SerialDataReceivedEventArgs e)
// อินเทอร์เน็ตยูอาร์ทเป็นการแปลงสัญญาณรูปแบบขนานจากคอมพิวเตอร์ให้
// เป็นรูปแบบอนุกรมแบบอะซิงโครนัส(พอร์ตอนุกรม RS 232)
{
inData = comPort.ReadExisting().Trim();
Data_DataReceived(); // เรียกใช้ฟังก์ชันข้อมูล คาดัวีซีฟว

```

```

}
private void Data_DataReceived() // คาดำรีชีฟว'ฟังก์ชัน
{
intData = Convert.ToInt32(inData);
if (intData<= 110 &&intData>= 10) // ค่าจริงที่หน้าต่าง โปรแกรมสามารถ
มองเห็นคือ ตั้งแต่10 ถึง 110
{
tB1.Text = inData;
}
if (intData< 10 &&intData>= 0) // ค่าจริงที่น้อยกว่า 10 หน้าต่างจะโชว์
เป็น0
{
tB1.Text = "0";
}
}
private void btSendData_Click(object sender, EventArgs e) // คลิก ปุ่ม "ส่งค่า"
{
intintInput;
intInput = Convert.ToInt32(tB2.Text);
if (comPort.IsOpen) { comPort.Write(intInput.ToString()+"#");
}
}
private void bt_noair_Click(object sender, EventArgs e) // คลิก ปุ่ม "ยางแบน"
{
if (comPort.IsOpen)
{
comPort.Write("n");
}
}
private void btAotoMan_Click(object sender, EventArgs e) // คลิกปุ่ม
แมนนำวล/ออก
ได้

```

```

{
if (mode == 1) // เริ่มต้นโหมดจะเป็น 1
{
mode = 0; // เมื่อคลิกปุ่ม โหมดจะกลายเป็น 0
cbCar.Enabled = false; //
cbComp.Enabled = false; //
tB2.ReadOnly = false; //
rb1.Enabled = false; //
rb2.Enabled = false; // ปุ่มต่างๆ ฝั่งออกได้จะใช้งานไม่ได้
cbCar.Text = ""; //
cbComp.Text = ""; //
tB2.Text = ""; //
rb1.Checked = false; //
rb2.Checked = false; //
}
else
{
mode = 1; // เมื่อคลิกปุ่ม โหมดจะกลายเป็น 1
cbCar.Enabled = true; //
cbComp.Enabled = true; //
tB2.ReadOnly = true; // ปุ่มต่างๆ ฝั่งออกได้จะใช้งานได้ปกติ
rb1.Enabled = true; //
rb2.Enabled = true; //
}
}

private void btSelecCar_Click(object sender, EventArgs e) // ในส่วนของปุ่มบันทึก
// ทอน(ปุ่มเลือก) เพื่อที่จะส่งค่าไปที่
// แท็บที่สอง(ช่องตั้งค่า)
{
stringselectedCar = cbCar.Items[cbCar.SelectedIndex].ToString();

/***** ฐิติ *****/

```

```

if (selectedCar == "CITY")
{
if (rb1.Checked)
{
value_car_auto = 27;           // ล้อหน้า
}
else if (rb2.Checked)
{
value_car_auto = 27;           // ล้อหลัง
}
else { value_car_auto = 0;     // ไม่เลือก
}
}
/*****/
/*****  Civic *****/
else if (selectedCar == "CIVIC")
{
if (rb1.Checked)
{
value_car_auto = 30;           // ล้อหน้า
}
else if (rb2.Checked)
{
value_car_auto = 29;           // ล้อหลัง
}
else
{
value_car_auto = 0;           // ไม่เลือก
}
}
/*****/

```

```

/***** แอคคอร์ด *****/
else if (selectedCar == "ACCORD")
{
if (rb1.Checked)
{
value_car_auto = 33;           // ล้อหน้า
}
else if (rb2.Checked)
{
value_car_auto = 32;           // ล้อหลัง
}
else
{
value_car_auto = 0;           // ไม่เลือก
}
}
/***** วีออส *****/
else if (selectedCar == "VIOS")
{
if (rb1.Checked)
{
value_car_auto = 32;           // ล้อหน้า
}
else if (rb2.Checked)
{
value_car_auto = 30;           // ล้อหลัง
}
else
{
value_car_auto = 0;           // ไม่เลือก
}
}

```

```

}

/*****
/***** วิโก้ *****/

else if (selectedCar == "VIGO")
{
if (rb1.Checked)
{
value_car_auto = 32;           // ล้อหน้า
}

else if (rb2.Checked)
{
value_car_auto = 34;           // ล้อหลัง
}

else
{
value_car_auto = 0;           // ไม่เลือก
}
}

/*****
/***** เคนารี *****/

else if (selectedCar == "CAMRY")
{
if (rb1.Checked)
{
value_car_auto = 32;           // ล้อหน้า
}

else if (rb2.Checked)
{
value_car_auto = 29;           // ล้อหลัง
}

else
{

```

```

value_car_auto = 0;           // ไม่เลือก
}
}

/*****
/***** ซันนี่ *****/

else if (selectedCar == "SUNNY")
{
if (rb1.Checked)
{
value_car_auto = 32;       // ล้อหน้า
}
else if (rb2.Checked)
{
value_car_auto = 29;       // ล้อหลัง
}
else
{
value_car_auto = 0;        // ไม่เลือก
}
}

/*****
/***** เทน้า *****/

else if (selectedCar == "TEANA")
{
if (rb1.Checked)
{
value_car_auto = 32;       // ล้อหน้า
}
else if (rb2.Checked)
{
value_car_auto = 30;       // ล้อหลัง
}
}

```



```

else
{
value_car_auto = 0;           // ไม่เลือก
}
}

/*****
/***** มาร์ช *****/
else if (selectedCar == "MARCH")
{
if (rb1.Checked)
{
value_car_auto = 32;         // ล้อหน้า
}
else if (rb2.Checked)
{
value_car_auto = 32;         // ล้อหลัง
}
else
{
value_car_auto = 0;         // ไม่เลือก
}
}

/*****
/***** เคนเซอร์ *****/
else if (selectedCar == "LANCER")
{
if (rb1.Checked)
{
value_car_auto = 29;         // ล้อหน้า
}
else if (rb2.Checked)
{

```

```

value_car_auto = 27;           // ล้อหลัง
}
else
{
value_car_auto = 0;           // ไม่เลือก
}
}

/*****
/***** แอทเรนเจอร์ *****/

else if (selectedCar == "ATTRANGER")
{
if (rb1.Checked)
{
value_car_auto = 30;         // ล้อหน้า
}
else if (rb2.Checked)
{
value_car_auto = 30;         // ล้อหลัง
}
else
{
value_car_auto = 0;         // ไม่เลือก
}
}

/*****
/***** ปาเจโร่ *****/

else if (selectedCar == "PAJERO")
{
if (rb1.Checked)
{
value_car_auto = 32;         // ล้อหน้า
}
}

```

```
else if (rb2.Checked)
{
value_car_auto = 36;           // ถ้อยหลัง
}
else
{
value_car_auto = 0;           // ไม่เลือก
}
}
else{}
/*****/
/*****/
tB2.Text = value_car_auto.ToString();
}
}
}
```





ภาคผนวก ก

รายละเอียดของไมโครคอนโทรลเลอร์ PIC 16F877



MICROCHIP

PIC16F87X
Data Sheet

28/40-Pin 8-Bit CMOS FLASH

Microcontrollers





MICROCHIP

PIC16F87X

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

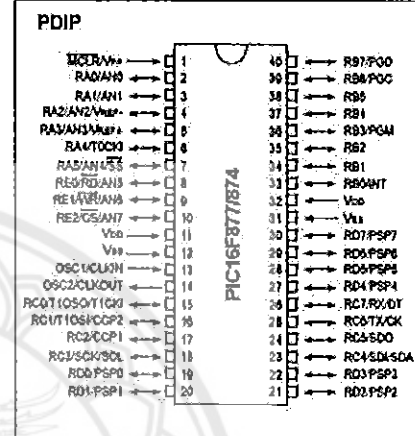
Devices Included in this Data Sheet:

- PIC16F873
- PIC16F876
- PIC16F874
- PIC16F877

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,
Up to 368 x 8 bytes of Data Memory (RAM)
Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, Indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and
Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC
oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM
technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two
pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature
ranges
- Low-power consumption:
 - < 0.6 mA typical @ 3V, 4 MHz
 - 20 µA typical @ 3V, 32 kHz
 - < 1 µA typical standby current

Pin Diagram

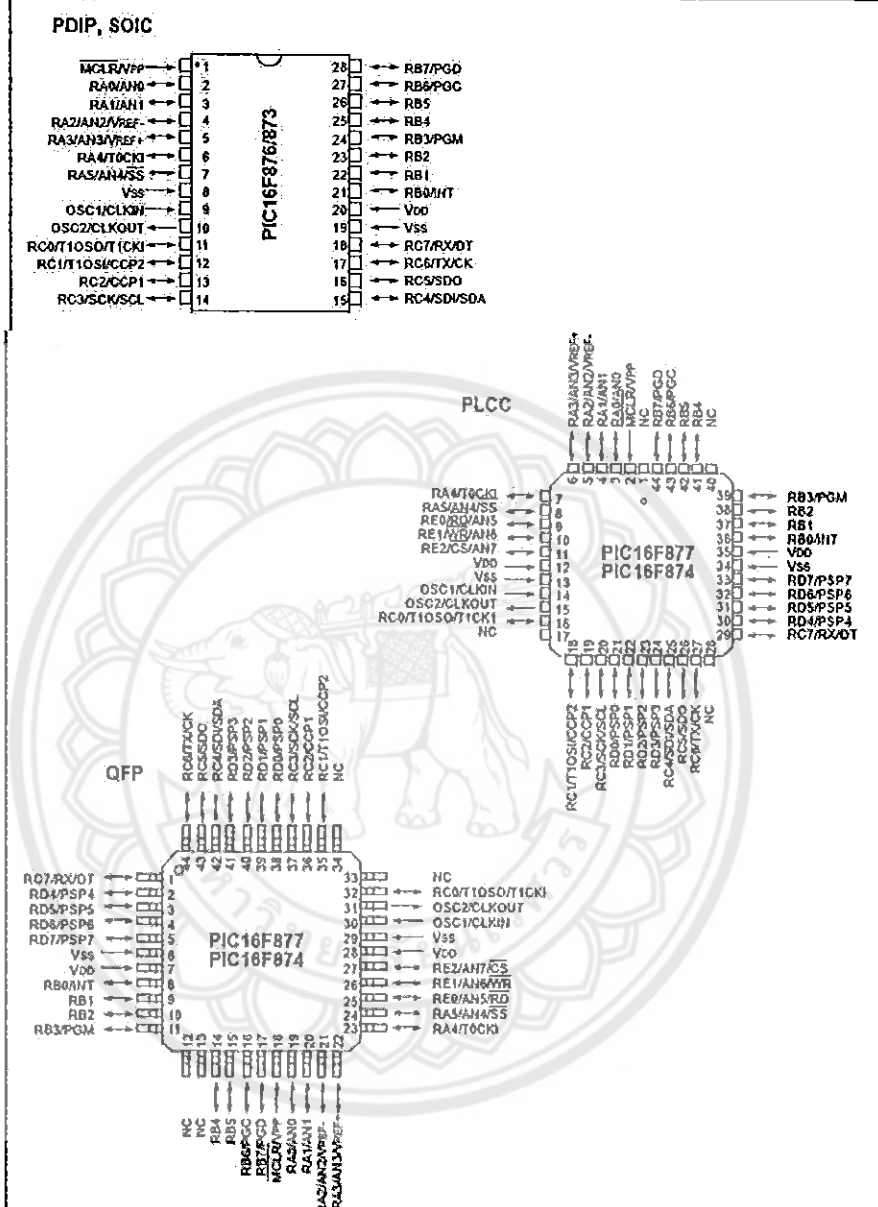


Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler,
can be incremented during SLEEP via external
crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period
register, prescaler and postscaler
- Two Capture, Compare, PWM modules
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master
mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver
Transmitter (USART/SCI) with 9-bit address
detection
- Parallel Slave Port (PSP) 8-bits wide, with
external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for
Brown-out Reset (BOR)

PIC16F87X

Pin Diagrams

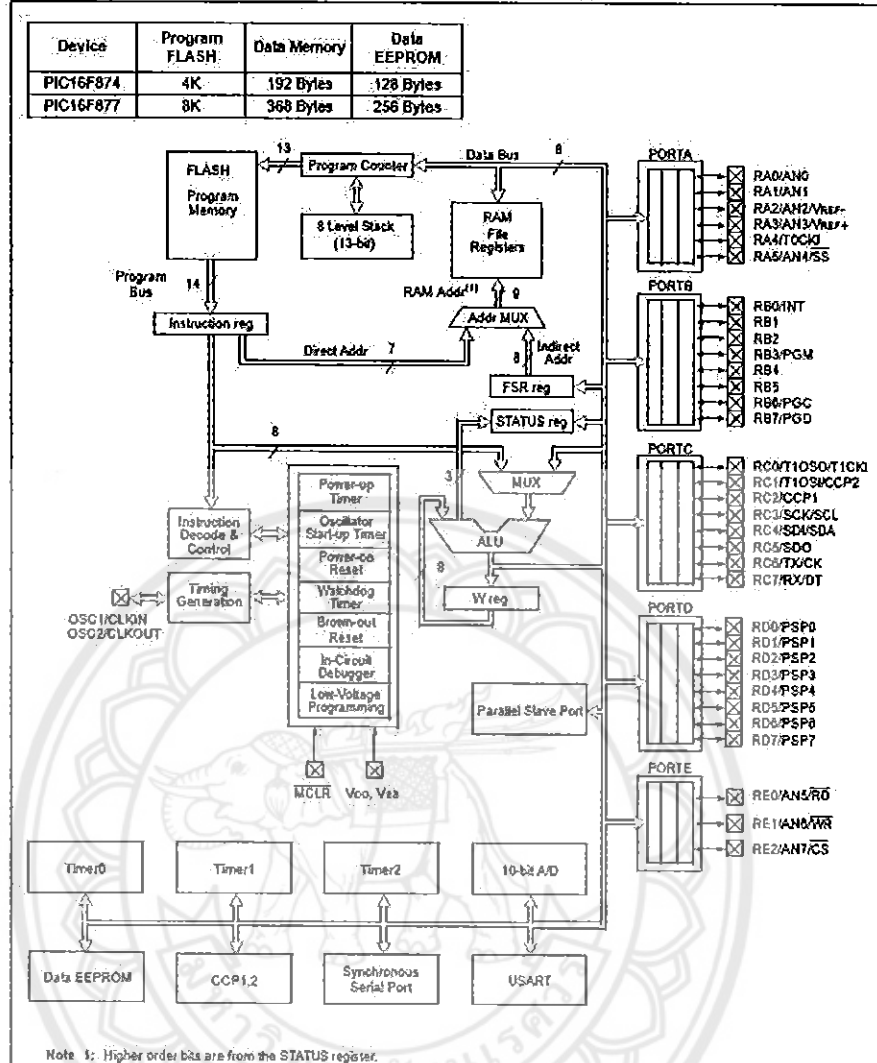


PIC16F87X

Key Features PICmicro™ Mid-Range Reference Manual (DS33023)	PIC16F873	PIC16F874	PIC16F876	PIC16F877
Operating Frequency	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz
RESETS (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
FLASH Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory	128	128	256	256
Interrupts	13	14	13	14
I/O Ports	Ports A,B,C	Ports A,B,C,D,E	Ports A,B,C	Ports A,B,C,D,E
Timers	3	3	3	3
Capture/Compare/PWM Modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Instruction Set	35 instructions	35 instructions	35 instructions	35 instructions



FIGURE 1-2: PIC16F874 AND PIC16F877 BLOCK DIAGRAM



PIC16F87X

3.0 I/O PORTS

Some pins for these I/O ports are multiplexed with an alternate function for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

Additional information on I/O ports may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023).

3.1 PORTA and the TRISA Register

PORTA is a 6-bit wide, bi-directional port. The corresponding data direction register is TRISA. Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., put the corresponding output driver in a Hi-Impedance mode). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., put the contents of the output latch on the selected pin).

Reading the PORTA register reads the status of the pins, whereas writing to it will write to the port latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, the value is modified and then written to the port data latch.

Pin RA4 is multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin. The RA4/T0CKI pin is a Schmitt Trigger input and an open drain output. All other PORTA pins have TTL input levels and full CMOS output drivers.

Other PORTA pins are multiplexed with analog inputs and analog VREF input. The operation of each pin is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register1).

Note: On a Power-on Reset, these pins are configured as analog inputs and read as '0'.

The TRISA register controls the direction of the RA pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

EXAMPLE 3.1: INITIALIZING PORTA

```
BCF STATUS, RP0 ;
BCF STATUS, RP1 ; Bank0
CLRF PORTA ; Initialize PORTA by
; clearing output
; data latches
BSF STATUS, RP0 ; Select Bank 1
MOVLW 0x06 ; Configure all pins
MOVWF ADCON1 ; as digital inputs
MOVLW 0x0F ; Value used to
; initialize data
; direction
MOVWF TRISA ; Set RA<3:0> as inputs
; RA<5:4> as outputs
; TRISA<7:6> are always
; read as '0'.
```

FIGURE 3-1: BLOCK DIAGRAM OF RA3:RA0 AND RA5 PINS

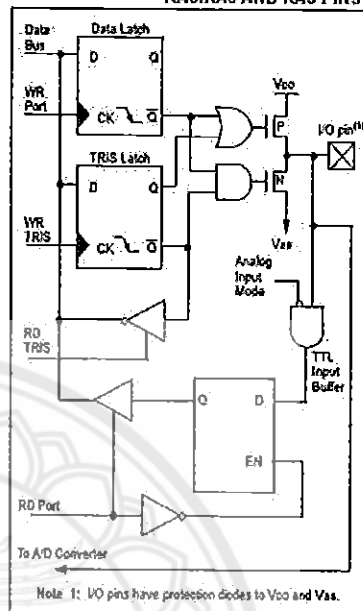
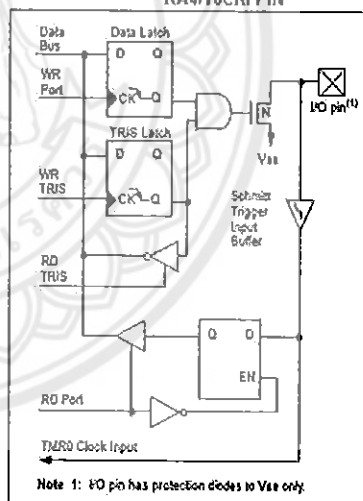


FIGURE 3-2: BLOCK DIAGRAM OF RA4/T0CKI PIN



PIC16F87X

TABLE 3-1: PORTA FUNCTIONS

Name	Bit#	Buffer	Function
RA0/AN0	bit0	TTL	Input/output or analog input.
RA1/AN1	bit1	TTL	Input/output or analog input.
RA2/AN2	bit2	TTL	Input/output or analog input.
RA3/AN3/VREF	bit3	TTL	Input/output or analog input or VREF.
RA4/TOCKI	bit4	ST	Input/output or external clock input for Timer0. Output is open drain type.
RA5/SS/AN4	bit5	TTL	Input/output or slave select input for synchronous serial port or analog input.

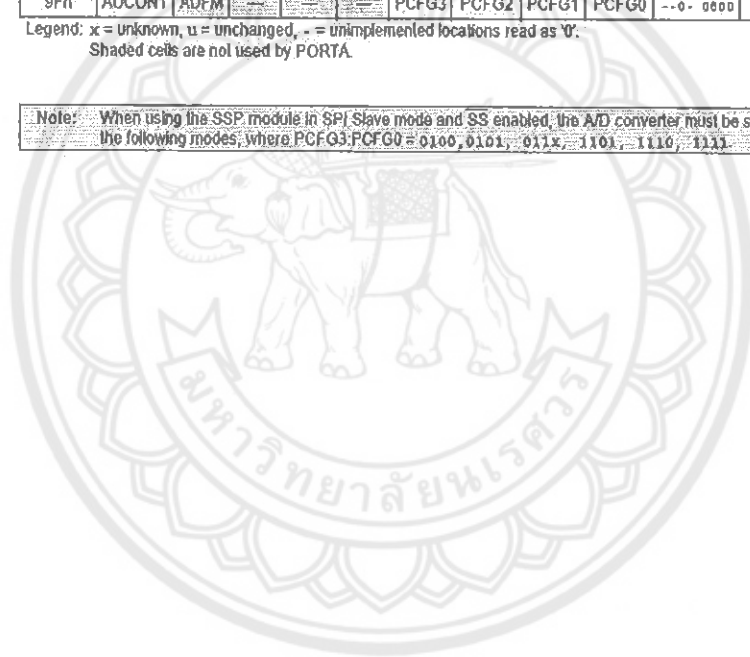
Legend: TTL = TTL Input; ST = Schmitt Trigger Input

TABLE 3-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
05h	PORTA			RA5	RA4	RA3	RA2	RA1	RA0	--0x 0000	--0x 0000
85h	TRISA			PORTA Data Direction Register						--11 1111	--11 1111
9Fh	ADCON1	ADFM				PCFG3	PCFG2	PCFG1	PCFG0	--0- 0000	--0- 0000

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'.
Shaded cells are not used by PORTA.

Note: When using the SSP module in SPI Slave mode and SS enabled, the A/D converter must be set to one of the following modes, where PCFG3:PCFG0 = 0100, 0101, 011x, 1101, 1110, 1111.



PIC16F87X

TABLE 3-3: PORTB FUNCTIONS

Name	Bit#	Buffer	Function
RB0/INT	bit0	TTL/ST ⁽¹⁾	Input/output pin or external interrupt input. Internal software programmable weak pull-up.
RB1	bit1	TTL	Input/output pin. Internal software programmable weak pull-up.
RB2	bit2	TTL	Input/output pin. Internal software programmable weak pull-up.
RB3/PGM ⁽³⁾	bit3	TTL	Input/output pin or programming pin in LVP mode. Internal software programmable weak pull-up.
RB4	bit4	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB5	bit5	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB6/PGC	bit6	TTL/ST ⁽²⁾	Input/output pin (with interrupt-on-change) or In-Circuit Debugger pin. Internal software programmable weak pull-up. Serial programming clock.
RB7/PGD	bit7	TTL/ST ⁽²⁾	Input/output pin (with interrupt-on-change) or In-Circuit Debugger pin. Internal software programmable weak pull-up. Serial programming data.

Legend: TTL = TTL Input, ST = Schmitt Trigger Input

Note 1: This buffer is a Schmitt Trigger Input when configured as the external interrupt.

2: This buffer is a Schmitt Trigger Input when used in Serial Programming mode.

3: Low Voltage ICSP Programming (LVP) is enabled by default, which disables the RB3 I/O function. LVP must be disabled to enable RB3 as an I/O pin and allow maximum compatibility to the other 28-pin and 40-pin mid-range devices.

TABLE 3-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
06h, 106h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	x000x x000x	uuuu uuuu
86h, 186h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
81h, 181h	OPTION_REG	RBPU	INTE0G	TOGS	TOSE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: x = unknown, u = unchanged. Shaded cells are not used by PORTB.

PIC16F87X

TABLE 3-5: PORTC FUNCTIONS

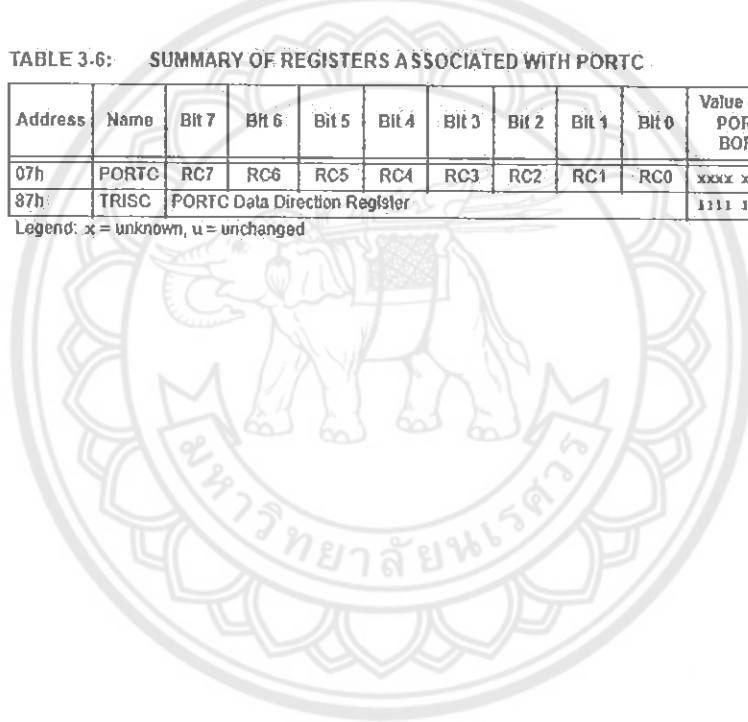
Name	Bit#	Buffer Type	Function
RC0/T1OSO/T1CKI	bit0	ST	Input/output port pin or Timer1 oscillator output/Timer1 clock input.
RC1/T1OSI/CCP2	bit1	ST	Input/output port pin or Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output.
RC2/CCP1	bit2	ST	Input/output port pin or Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL	bit3	ST	RC3 can also be the synchronous serial clock for both SPI and I ² C modes.
RC4/SDI/SDA	bit4	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I ² C mode).
RC5/SDO	bit5	ST	Input/output port pin or Synchronous Serial Port data output.
RC6/TX/CK	bit6	ST	Input/output port pin or USART Asynchronous Transmit or Synchronous Clock.
RC7/RX/DT	bit7	ST	Input/output port pin or USART Asynchronous Receive or Synchronous Data.

Legend: ST = Schmitt Trigger Input

TABLE 3-6: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	nnnn nnnn
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111

Legend: x = unknown, u = unchanged



PIC16F87X

3.4 PORTD and TRISD Registers

PORTD and TRISD are not implemented on the PIC16F873 or PIC16F876.

PORTD is an 8-bit port with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

PORTD can be configured as an 8-bit wide microprocessor port (parallel slave port) by setting control bit PSPMODE (TRISE<4>). In this mode, the input buffers are TTL.

FIGURE 3-7: PORTD BLOCK DIAGRAM (IN I/O PORT MODE)

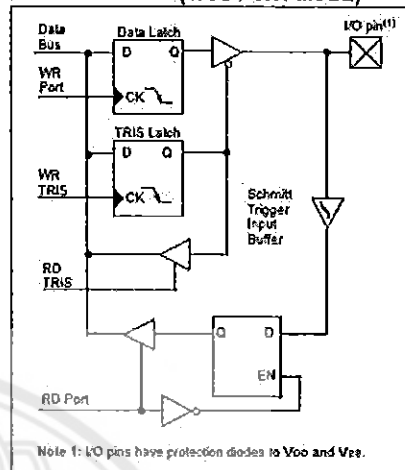


TABLE 3-7: PORTD FUNCTIONS

Name	Bit#	Buffer Type	Function
RD0/PSP0	bit0	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit0.
RD1/PSP1	bit1	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit1.
RD2/PSP2	bit2	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit2.
RD3/PSP3	bit3	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit3.
RD4/PSP4	bit4	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit4.
RD5/PSP5	bit5	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit5.
RD6/PSP6	bit6	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit6.
RD7/PSP7	bit7	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit7.

Legend: ST = Schmitt Trigger input, TTL = TTL input

Note 1: Input buffers are Schmitt Triggers when in I/O mode and TTL buffers when in Parallel Slave Port mode.

TABLE 3-8: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
08h	PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	0000 0000
88h	TRISD	PORTD Data Direction Register								1111 1111	1111 1111
89h	TRISE	IBF	OBF	IBOV	PSPMODE	PORTD Data Direction Bits				0000 -111	0000 -111

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by PORTD.

PIC16F87X

3.5 PORTE and TRISE Register

PORTE and TRISE are not implemented on the PIC16F873 or PIC16F876.

PORTE has three pins (RE0/RD/AN5, RE1/WR/AN6, and RE2/CS/AN7) which are individually configurable as inputs or outputs. These pins have Schmitt Trigger Input buffers.

The PORTE pins become the I/O control inputs for the microprocessor port when bit PSPMODE (TRISE<4>) is set. In this mode, the user must make certain that the TRISE<2:0> bits are set, and that the pins are configured as digital inputs. Also ensure that ADCON1 is configured for digital I/O. In this mode, the input buffers are TTL.

Register 3-1 shows the TRISE register, which also controls the parallel slave port operation.

PORTE pins are multiplexed with analog inputs. When selected for analog input, these pins will read as '0's.

TRISE controls the direction of the RE pins, even when they are being used as analog inputs. The user must make sure to keep the pins configured as inputs when using them as analog inputs.

Note: On a Power-on Reset, these pins are configured as analog inputs, and read as '0'.

FIGURE 3-8: PORTE BLOCK DIAGRAM (IN I/O PORT MODE)

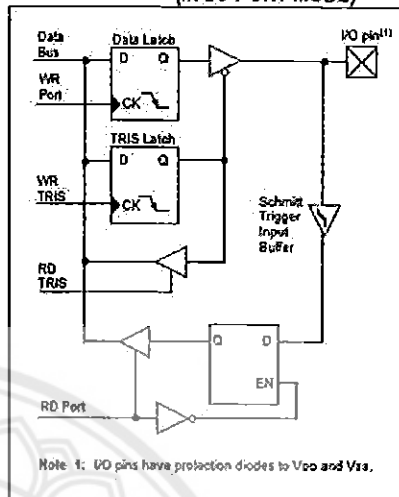


TABLE 3-9: PORTE FUNCTIONS

Name	Bit#	Buffer Type	Function
RE0/RD/AN5	bit0	ST/TTL ⁽¹⁾	I/O port pin or read control input in Parallel Slave Port mode or analog input: RD 1 = Idle 0 = Read operation. Contents of PORTD register are output to PORTD I/O pins (if chip selected)
RE1/WR/AN6	bit1	ST/TTL ⁽¹⁾	I/O port pin or write control input in Parallel Slave Port mode or analog input: WR 1 = Idle 0 = Write operation. Value of PORTD I/O pins is latched into PORTD register (if chip selected)
RE2/CS/AN7	bit2	ST/TTL ⁽¹⁾	I/O port pin or chip select control input in Parallel Slave Port mode or analog input: CS 1 = Device is not selected 0 = Device is selected

Legend: ST = Schmitt Trigger input, TTL = TTL input

Note 1: Input buffers are Schmitt Triggers when in I/O mode and TTL buffers when in Parallel Slave Port mode.

TABLE 3-10: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
09h	PORTE	—	—	—	—	—	RE2	RE1	RE0	— — — — — xxx	— — — — — uuu
89h	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits			0000 - 111	0000 - 111
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	— 0 - 0000	— 0 - 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by PORTE.

PIC16F87X

10.1 USART Baud Rate Generator (BRG)

The BRG supports both the Asynchronous and Synchronous modes of the USART. It is a dedicated 8-bit baud rate generator. The SPBRG register controls the period of a free running 8-bit timer. In Asynchronous mode, bit BRGH (TXSTA<2>) also controls the baud rate. In Synchronous mode, bit BRGH is ignored. Table 10-1 shows the formula for computation of the baud rate for different USART modes which only apply in Master mode (internal clock).

Given the desired baud rate and F_{osc} , the nearest integer value for the SPBRG register can be calculated using the formula in Table 10-1. From this, the error in baud rate can be determined.

It may be advantageous to use the high baud rate (BRGH = 1), even for slower baud clocks. This is because the $F_{osc}/(16(X+1))$ equation can reduce the baud rate error in some cases.

Writing a new value to the SPBRG register causes the BRG timer to be reset (or cleared). This ensures the BRG does not wait for a timer overflow before outputting the new baud rate.

10.1.1 SAMPLING

The data on the RC7/RX/DT pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX pin.

TABLE 10-1: BAUD RATE FORMULA

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = $F_{osc}/(64(X+1))$	Baud Rate = $F_{osc}/(16(X+1))$
1	(Synchronous) Baud Rate = $F_{osc}/(4(X+1))$	N/A

X = value in SPBRG (0 to 255)

TABLE 10-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
98h	TXSTA	CSRC	TX9	TXEN	SYNC		BRGH	TRMT	TX9D	0000 - 010	0000 - 010
16h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used by the BRG.



ภาคผนวก ง

รายละเอียดเกี่ยวกับ MPX 5700

**Integrated Silicon Pressure Sensor
On-Chip Signal Conditioned,
Temperature Compensated and
Calibrated**

The MPX5700 series piezoresistive transducer is a state-of-the-art monolithic silicon pressure sensor designed for a wide range of applications, but particularly those employing a microcontroller or microprocessor with A/D inputs. This patented, single element transducer combines advanced micromachining techniques, thin-film metallization, and bipolar processing to provide an accurate, high level analog output signal that is proportional to the applied pressure.

Features

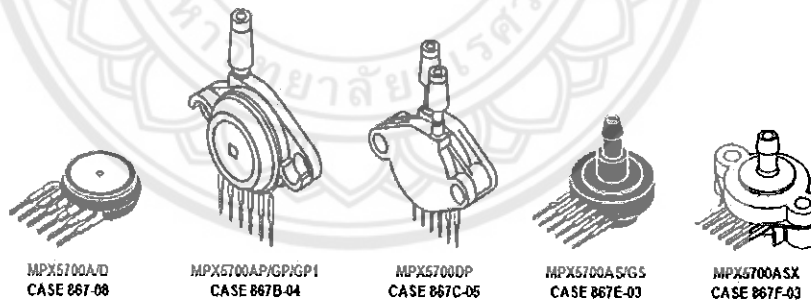
- 2.5% Maximum Error over 0° to 85°C
- Ideally Suited for Microprocessor or Microcontroller-Based Systems
- Available in Absolute, Differential and Gauge Configurations
- Patented Silicon Shear Stress-Strain Gauge
- Durable Epoxy Unibody Element

**MPX5700
Series**

0 to 700 kPa (0 to 101.5 psi)
15 to 700 kPa (2.1 to 101.5 psi)
0.2 to 4.7 V output

Device Name	Case No.	# of Ports			Pressure Type			Device Name
		None	Single	Dual	Gauge	Differential	Absolute	
Unibody Package (MPX5700 Series)								
MPX5700A	867	*					*	MPX5700A
MPX5700AP	867B		*				*	MPX5700AP
MPX5700AS	867E		*				*	MPX5700A
MPX5700ASX	867F		*				*	MPX5700A
MPX5700D	867	*				*		MPX5700D
MPX5700DP	867C			*		*		MPX5700DP
MPX5700GP	867B		*		*			MPX5700GP
MPX5700GP1 ⁽¹⁾	867B		*		*			MPX5700GP
MPX5700GS	867E		*		*			MPX5700D

1. MPX5700GP1 has 90 degree lead form.

UNIBODY PACKAGES

Pressure

Operating Characteristics

Table 1. Operating Characteristics ($V_S = 5.0$ Vdc, $T_A = 25^\circ\text{C}$ unless otherwise noted, $P_1 > P_2$. Decoupling circuit shown in required to meet electrical specifications.)

Characteristic	Symbol	Min	Typ	Max	Unit
Pressure Range ⁽¹⁾ Gauge, Differential: MPX5700D Absolute: MPX5700A	P_{OP}	0 15	—	700 700	kPa
Supply Voltage ⁽²⁾	V_S	4.75	5.0	5.25	Vdc
Supply Current	I_O	—	7.0	10	mAdc
Zero Pressure Offset ⁽³⁾ Gauge, Differential (0 to 85°C) Absolute (0 to 85°C)	V_{OFF}	0.088 0.184	0.2	0.313 0.409	Vdc
Full Scale Output ⁽⁴⁾ (0 to 85°C)	V_{FSO}	4.587	4.7	4.819	Vdc
Full Scale Span ⁽⁵⁾ (0 to 85°C)	V_{FSS}	—	4.5	—	Vdc
Accuracy ⁽⁶⁾ (0 to 85°C)	—	—	—	±2.5	% V_{FSS}
Sensitivity	V/P	—	6.4	—	mV/kPa
Response Time ⁽⁷⁾	t_R	—	1.0	—	ms
Output Source Current at Full Scale Output	I_{OS}	—	0.1	—	mAdc
Warm-Up Time ⁽⁸⁾	—	—	20	—	ms

1. 1.0 kPa (kiloPascal) equals 0.145 psi.
2. Device is ratiometric within this specified excitation range.
3. Offset (V_{OFF}) is defined as the output voltage at the minimum rated pressure.
4. Full Scale Output (V_{FSO}) is defined as the output voltage at the maximum or full rated pressure.
5. Full Scale Span (V_{FSS}) is defined as the algebraic difference between the output voltage at full rated pressure and the output voltage at the minimum rated pressure.
6. Accuracy (error budget) consists of the following:
Linearity: Output deviation from a straight line relationship with pressure over the specified pressure range.
Temperature Hysteresis: Output deviation at any temperature within the operating temperature range, after the temperature is cycled to and from the minimum or maximum operating temperature points, with zero differential pressure applied.
Pressure Hysteresis: Output deviation at any pressure within the specified range, when this pressure is cycled to and from the minimum or maximum rated pressure, at 25°C.
TcSpan: Output deviation over the temperature range of 0° to 85°C, relative to 25°C.
TcOffset: Output deviation with minimum rated pressure applied, over the temperature range of 0° to 85°C, relative to 25°C.
Variation from Nominal: The variation from nominal values, for Offset or Full Scale Span, as a percent of V_{FSS} , at 25°C.
7. Response Time is defined as the time for the incremental change in the output to go from 10% to 90% of its final value when subjected to a specified step change in pressure.
8. Warm-up Time is defined as the time required for the device to meet the specified output voltage after the pressure has been stabilized.

Pressure

Maximum Ratings

Table 2. Maximum Ratings⁽¹⁾

Parameter	Symbol	Value	Unit
Maximum Pressure ⁽²⁾ (P2 ≤ 1 Atmosphere)	P1 _{max}	2800	kPa
Storage Temperature	T _{stg}	-40 to +125	°C
Operating Temperature	T _A	-40 to +125	°C

1. Maximum Ratings apply to Case 867 only. Extended exposure at the specified limits may cause permanent damage or degradation to the device.
2. This sensor is designed for applications where P1 is always greater than, or equal to P2. P2 maximum is 500 kPa.

Figure 1 shows a block diagram of the internal circuitry integrated on a pressure sensor chip.

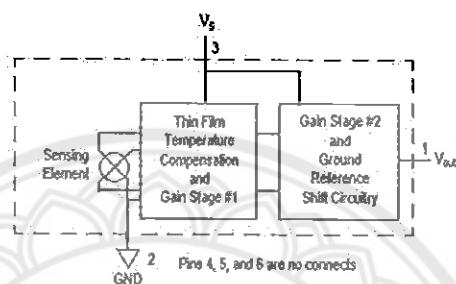


Figure 1. Fully Integrated Pressure Sensor Schematic

Pressure

On-chip Temperature Compensation and Calibration

Figure 3 illustrates both the Differential/Gauge and the Absolute Sensing Chip in the basic chip carrier (Case 867). A fluorosilicone gel isolates the die surface and wire bonds from the environment, while allowing the pressure signal to be transmitted to the sensor diaphragm. (For use of the MPX5700D in a high-pressure cyclic application, consult the factory.)

The MPX5700 series pressure sensor operating characteristics, and internal reliability and qualification tests are based on use of dry air as the pressure media. Media, other than dry air, may have adverse effects on sensor

performance and long-term reliability. Contact the factory for information regarding media compatibility in your application.

Figure 2 shows the sensor output signal relative to pressure input. Typical, minimum, and maximum output curves are shown for operation over a temperature range of 0° to 85°C using the decoupling circuit shown in . The output will saturate outside of the specified pressure range.

shows the recommended decoupling circuit for interfacing the output of the integrated sensor to the A/D input of a microprocessor or microcontroller. Proper decoupling of the power supply is recommended.

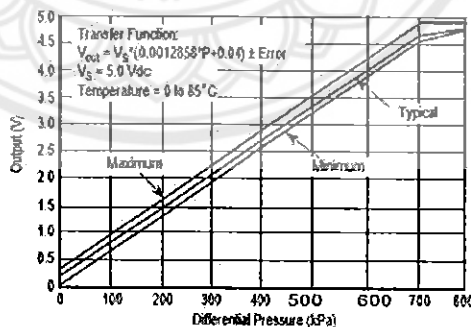


Figure 2. Output vs. Pressure Differential

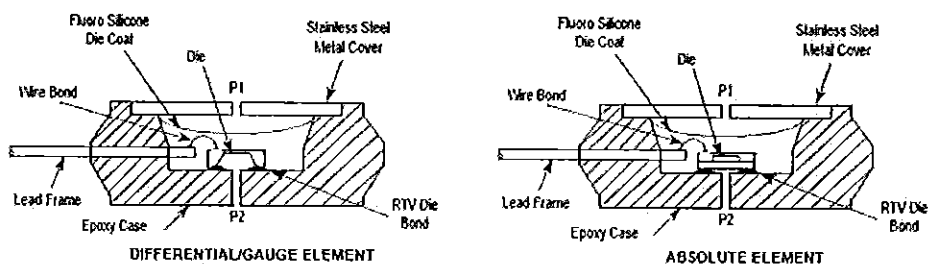


Figure 3. Cross-Sectional Diagrams (not to scale)

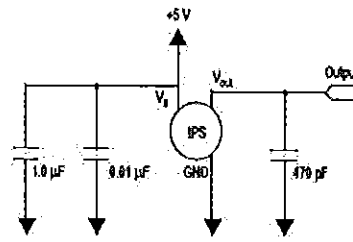


Figure 4. Recommended Power Supply Decoupling and Output Filtering
(For additional output filtering, please refer to Application Note AN1646)

MPX5700

4

Sensors
Freescale Semiconductor, Inc.

Pressure

PRESSURE (P1)/VACUUM (P2) SIDE IDENTIFICATION TABLE

Freescale designates the two sides of the pressure sensor as the Pressure (P1) side and the Vacuum (P2) side. The Pressure (P1) side is the side containing fluorosilicone gel which protects the die from harsh media. The Freescale MPX

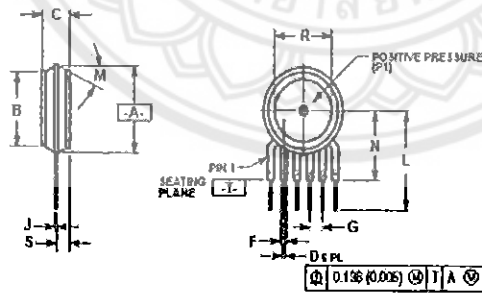
pressure sensor is designed to operate with positive differential pressure applied, P1 > P2.

The Pressure (P1) side may be identified by using the following table.

Part Number	Case Type	Pressure (P1) Side Identifier
MPX5700AD	867	Stainless Steel Cap
MPX5700DP	867C	Side with Part Marking
MPX5700GP/AP	867B	Side with Port Attached
MPX5700GS/AS	867E	Side with Port Attached
MPX5700ASX	867F	Side with Port Attached

Pressure

PACKAGE DIMENSIONS



- NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.1M, 1992.
 2. CONTROLLING DIMENSION: INCH.
 3. DIMENSION A IS INCLUSIVE OF THE WELD STOP RING, WELD STOP RING NOT TO EXCEED 0.005 (0.020).

DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.365	0.520	9.27	13.20
B	0.511	0.524	12.96	13.30
C	0.208	0.228	5.28	5.79
D	0.087	0.092	2.20	2.34
F	0.098	0.094	2.50	2.40
G	0.100 BSC			
J	0.094	0.094	2.39	2.39
L	0.085	0.125	2.16	3.18
M	SEE NOTE 3			
N	0.425	0.465	10.80	11.81
P	0.400	0.450	10.16	11.43
S	0.084	0.100	2.13	2.54

- STR1:1
PW1: VOI
2: GND
3: VCC
4: V1
5: V2
6: VEX

- STR1:7
PW1: OPEN
2: GND
3: VOI
4: SUPPLY
5: (VEX)
6: OPEN

- STR1:3
PW1: OPEN
2: GND
3: VOI
4: SUPPLY
5: VOI
6: OPEN

CASE 867-08
ISSUE N
BASIC ELEMENT

ประวัติผู้ดำเนินโครงการ



ชื่อ นายเรนศ สามลี

ภูมิลำเนา 150 หมู่ 3 ต. สบง อ. เมือง จ. พะเยา 56110

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจาก โรงเรียนภูซางวิทยาคม
- ปัจจุบันกำลังศึกษาระดับปริญญาตรี ชั้นปีที่ 4 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : hatorikung_top@hotmail.com



ชื่อ นายสุภวิชญ์ สังข์ยก

ภูมิลำเนา 1503/2 หมู่ 1 ต.สลกบาตร อ.ขามเฒ่าลักษ์บุรี จ.กำแพงเพชร 62140

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจาก โรงเรียนขามเฒ่าวิทยา
- ปัจจุบันกำลังศึกษาระดับปริญญาตรี ชั้นปีที่ 4 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : icecool_c99@hotmail.com